

N° d'enregistrement
au C.N.R.S.

THESE DE DOCTEUR INGENIEUR
MATHEMATIQUES
OPTION INFORMATIQUE

présentée

A LA FACULTE DES SCIENCES
de PARIS

par

Monsieur M.S. MANASFI

pour obtenir

le grade de Docteur Ingénieur

Sujet de la Thèse : Correction par ordinateur des altérations phonétiques des
noms français;

Calcul d'adresses et recherche en table.

Soutenue le : 18 Décembre 1970, à 14 h. devant la Commission d'Examen.

Président : Monsieur J. C. SIMON

Examineurs : Monsieur J. VIGNES
Monsieur C. PICARD

Invité : Monsieur M. ROCHER

THESE DE DOCTEUR INGENIEUR
MATHEMATIQUES
OPTION INFORMATIQUE

présentée

A LA FACULTE DES SCIENCES
de PARIS

par

Monsieur M.S. MANASFI

pour obtenir

le grade de Docteur Ingénieur

Sujet de la Thèse : Correction par ordinateur des altérations phonétiques des
noms français;

Calcul d'adresses et recherche en table.

Soutenue le : 18 Décembre 1970, à 14 h. devant la Commission d'Examen.

Président : Monsieur J. C. SIMON

Examineurs : Monsieur J. VIGNES
Monsieur C. PICARD

Invité : Monsieur M. ROCHER



Je tiens à exprimer mes plus vifs remerciements à Monsieur le Professeur J.C. SIMON pour m'avoir fait l'honneur de présider le Jury de cette thèse.

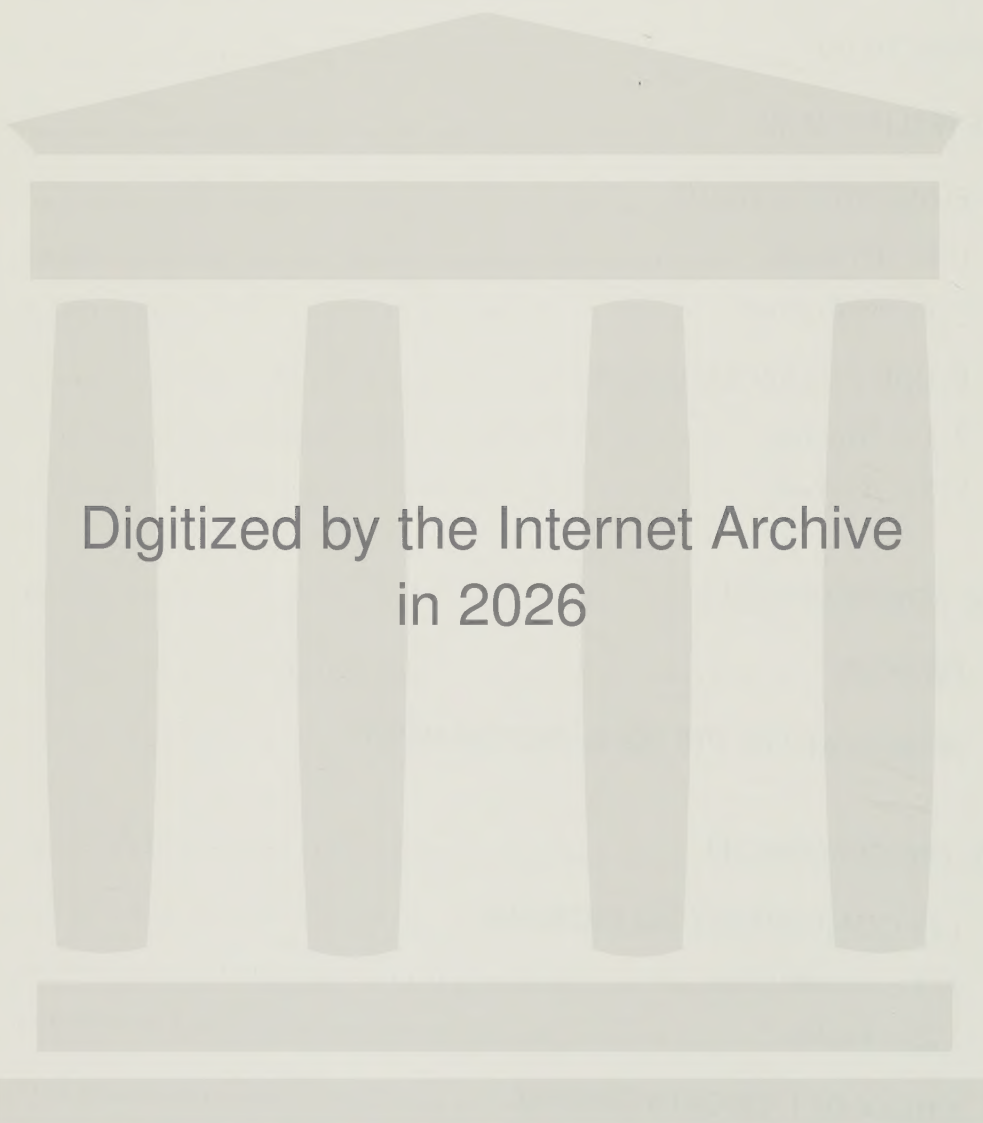
Je remercie également MM. les Professeurs J. VIGNES, C. PICARD et M. ROCHER d'avoir bien voulu faire partie du Jury.

J'adresse mes plus sincères remerciements à Monsieur le Directeur R.MOREAU pour toutes les facilités qu'il a mises à ma disposition au sein de son service à IBM France.

Que Monsieur P. PERSUY trouve ici l'expression de ma profonde gratitude pour les conseils et l'encouragement qu'il n'a cessé de me prodiguer pendant la préparation de ce travail

TABLE DES MATIERES

	Page
TABLE DES MATIERES	III
INTRODUCTION	1
ETUDE PRELIMINAIRE	3
1 - ETUDE PHONETIQUE	3
1.1 - Principe	3
1.2 - Réalisation	4
2 - ETUDE SYLLABOLOGIQUE	6
2.1 - Principe	6
2.2 - Analyse	7
ETUDE FONDAMENTALE	9
1 - PRINCIPE	9
2 - PRESENTATION DES SOUS-PROGRAMMES	10
SOUS-PROGRAMME SPI	12
1 - LES CONTRAINTES DU PROBLEME	12
1.1 - Introduction	12
1.2 - Règles	12
2 - CHOIX DE L'ORGANIGRAMME	15
3 - FONCTION DECOUPAGE	16
4 - ELIMINATION DES E MUETS	17
5 - COMPTAGE	21



Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41551520_0013

	Page
SOUS-PROGRAMME SP2	23
1 - LES STRUCTURES DES TABLES	23
1.1 - Structure	23
1.1.1 - Structure interne de l'IBM 360	23
1.1.2 - Structure des tables	24
2 - FONCTIONNEMENT	26
3 - ANALYSE	28
3.1 - Traitement du l	28
3.2 - Traitement des consonnes	30
SOUS-PROGRAMME SP3	33
1 - ELIMINATION DES CONSONNES MUETTES	33
2 - REDUCTION DES GROUPES VOCALIQUES	34
SOUS-PROGRAMME SP4	37
1 - ETUDE DE PRINCIPE	37
1.1 - Modificateurs	39
2 - ORGANIGRAMME	40
2.1 - Syllabe à un caractère	40
2.2 - Syllabe à deux caractères	41
2.2.1 - Cas . CV	41
2.2.2 - Cas . VC	41
2.3 - Syllabe à trois caractères	42

	Page
CHOIX DU CODE	45
1 - CRITERES	45
2 - CODAGE	46
2.1 - Classes de phonèmes	47
2.2 - Sous-classes	47
3 - MODIFICATEURS	49
3.1 - Voisinage	49
3.2 - Optimisation	51
3.3 - Codage des modificateurs	54
4 - ARRANGEMENT	59
RECHERCHE EN TABLE PAR LES FONCTIONS G	61
1 - RAPPEL	61
2 - ETUDE PRATIQUE	62
3 - OBSERVATIONS	62
4 - SOLUTION PROPOSEE	66
ETUDE PROBABILISTE	69
1 - INTRODUCTION	69
2 - PRINCIPE	70
2.1 - Matrice 27 x 27	72
2.2 - Exploitation de la matrice	74
3 - ALGORITHME	75
4 - OBSERVATIONS ET CONCLUSIONS	79
ANNEXE	85
BIBLIOGRAPHIE	86

INTRODUCTION

La précision et la rigueur avec lesquelles un ordinateur traite une information, font de ce dernier un outil de travail d'une performance sans égale, surtout dans la résolution mathématique des différents problèmes qui se posent à l'homme, où rigueur et précision sont les éléments fondamentaux pour l'aboutissement de tout raisonnement logique.

Dans ce domaine, l'ordinateur est un serviteur fidèle qui n'admet que les données logiques. Sa rigueur est tellement imposante qu'on le traite d'outil inhumain. Or, à notre échelle, l'erreur est une chose courante, nos sens ne sont pas parfaits, d'où l'extrême difficulté de tout dialogue avec une machine dont le taux d'erreur est réduit à un niveau infiniment petit.

L'étude proposée dans ces pages a pour but de rendre nos erreurs logiques, pour qu'elles soient acceptées par cet outil performant. Il est donc utile pour faire face aux problèmes traitants un gros volume de données de procéder à un certain rangement logique.

Quel que soit le support qui reçoit ces données, on est amené à créer un moyen de reconnaissance. Pour les uns c'est un numéro matricule, c'est-à-dire, un code numérique convenablement choisi. Pour d'autres, c'est un nom de marque ou un nom propre. Tout dépend de la nature du rangement et du fichier.

Dans beaucoup de cas, les noms propres de personnes sont utilisés comme mots-clés, permettant un accès direct à un enregistrement. Citons le cas de voyageurs d'avion, le cas de personnes adhérant à la Sécurité Sociale etc...

Un nom propre est constitué de groupements de mots, ayant des origines très variés. Un nom n'obéit pas dans sa conception à une suite logique de caractères, il est donc exposé à des altérations et des modifications dûs, parfois, à une mauvaise transcription du nom, ou bien à une mauvaise prononciation. On cite ici l'exemple type qui a servi comme base à notre étude; à savoir : les altérations phonétiques des noms propres transmis par téléphone. Si ce nom présente une légère modification, rien qu'une lettre à la place d'une autre ou bien une consonne muette manquante, il devient impossible de retrouver l'enregistrement correspondant.

Il serait donc souhaitable, à défaut de trouver l'enregistrement voulu, de faire admettre à l'ordinateur une certaine marge d'erreur, de tel façon qu'il puisse retrouver non pas l'enregistrement en question, mais tous les enregistrements qui lui ressemblent, conformément à des critères établis à l'avance.

Ces critères seront analysés et présentés au cours de cette étude.

ETUDE PRELIMINAIRE

1

ETUDE PHONETIQUE

1.1 - PRINCIPE

Inspiré des études citées en bibliographie, on a essayé d'aborder le problème par une méthode purement phonétique, c'est-à-dire, qu'on a essayé d'établir une liste de phonèmes qui se présentent et d'assigner à chaque phonème ou groupe de phonèmes une valeur déterminée. On était amené en sorte d'établir une correspondance biunivoque entre des sons et des symboles.

EXEMPLE

Si on donne aux phonèmes suivants les valeurs :

6	pour	ON
1	pour	IN
Ø	pour	EAU
8	pour	AI

on obtient comme équivalents des noms suivants :

BEAUMARCHAIS	→	BØMARCH2S
DUPONT	→	DUP5T
MARTIN	→	MART6

1.2 - REALISATION

On a réalisé un programme phonétique qui transcrit un nom donné en son équivalent phonétique. On a souvent donné des lettres équivalents aux phonèmes simples pour mieux voir les résultats dans un langage compréhensible. Un B, par exemple, a pour valeur un B tout simplement (Planche 1).

La réalisation du programme est basée sur la consultation de tables, résumant l'analyse des cas de groupement de phonèmes et de succession de lettres. Cette méthode qui sera exposée en détail plus loin, a servi pour mener à bonne fin notre étude.

La transcription phonétique pure ainsi effectuée, est hautement critiquable, et présente des insuffisances dont on peut citer comme suit :

- Il est indispensable d'avoir toutes les lettres présentes, pour avoir une transcription correcte. On suppose donc qu'il y a une équipartition de tous les phonèmes constituant un nom, ce qui n'est pas le cas.

Supposons que le R dans MARTIN soit omis, le nom se trouve donc tranqué d'une lettre au milieu. Les phonèmes subissent une translation vers la gauche, et toute recherche de ressemblance entre MARTIN et MATIN reste sans résultat.

- La transcription phonétique ne tient pas compte de l'emplacement des phonèmes, elle est basée sur la succession des lettres. Alors, qu'on sait très bien, qu'en prononçant on risque donc de masquer l'importance prise par des phonèmes privilégiés.
- La substitution d'une lettre par une autre peut provoquer des graves ennuis, même si la lettre substituée est assez proche de la lettre initiale.

Cette méthode connaît des applications d'un grand intérêt dans l'enseignement programmé (référence 7), ou bien dans un problème de créations de noms nouveaux pour produits pharmaceutiques ou commerciales (référence 10). Dans ces cas on doit décéler la présence de sons semblables en vue, soit de corriger, soit de créer de noms de marques sans risque de tomber dans la répétition.

2.1 - PRINCIPE

Une solution fort intéressante dans le traitement et la transmission automatique des textes de littérature est celle apportée par Monsieur MOREAU (référence 8).

Cette méthode propose le découpage en syllabes des mots au fur et à mesure qu'ils se présentent à l'ordinateur, moyennant quelques règles suivant lesquelles on autorise un découpage ou non, en se basant sur deux principes :

- La succession des lettres (consonnes/voyelles)
- Une table d'exception de groupement de deux ou plusieurs consonnes entre lesquelles il est interdit de couper un mot.

Le résultat obtenu est fort intéressant du moment où il présente un taux d'erreur très réduit.

Un exemple de ce découpage :

INTERESSANT → IN TE RES SANT

CONTRAIRE → CON TRAI RE

2.2 - ANALYSE

Le mode de découpage est grammatical. Il tient également compte de l'écriture pure et simple du nom. Il est fort intéressant dans les problèmes de codage de messages ou de télétransmission, mais n'amène pas une solution définitive à notre problème (voir planche 2).

EXEMPLE

Dans le mot IN TE RES SANT, le T final est muet, il n'a donc pas une valeur phonétique.

Le S dans RES est superflus, du moment où le son existe déjà dans SANT.

Dans le mot CON TRAI RE, le E final est muet, et on a une syllabe supplémentaire, le RE, car sans le E on peut rattacher R dans RE, à TRAI + R.

On voit donc que des deux méthodes citées plus haut, il serait souhaitable de trouver un moyen qui combine les avantages des deux à la fois et qui permet de maîtriser la variation du nom à n'importe quelle position et d'offrir un traitement unique et valable partout.

Bien sûr, notre étude est basée sur des données physiques pour la mise au point des différents programmes. On a utilisé un fichier constitué

de noms propres français et de certains noms propres francisés. Presque tous les éléments présentent des particularités d'écriture ou de prononciation. Toutefois, la méthode reste valable pour n'importe quelle langue latine avec de légères modifications.

ETUDE FONDAMENTALE

1

PRINCIPE

On s'est efforcé dans le programme principal, réalisant l'étude globale du problème, de séparer les différentes fonctions, en spécialisant des sous-programmes compilés à part "mode externe", en une ou deux fonctions.

Ce caractère modulaire que présente le programme, permettra par la suite, des modifications et des changements au niveau de la fonction elle-même, sans affecter le reste du traitement.

Le chaînage des sous-programmes étant assuré par une structure récursive, toutes les variables transmises entre les sous-programmes sont déclarées en externe. On réalise ainsi une grande liberté entre la fonction traitée qui représente une structure de squelette, et les variables sur lesquelles elle travaille.

Avant d'aborder l'étude détaillée des différents sous-programmes, résumons ici les différentes fonctions traitées par ces sous-programmes (SP).

1 - Sous-programme SP1

- Découpage du nom en syllabes phonétiques
- Elimination des E muets à l'intérieur et à la fin du nom
- Comptage statistique prédéterminant pour le choix du code.

2 - Sous-programme SP2

- Tables de consultation résumant toute l'analyse du problème.
- Codes.

3 - Sous-programme SP3

- Réduction des groupements vocaliques
- Elimination des consonnes muettes.

4 - Sous-programme SP4

- Réduction des syllabes - Homogénéisation.

5 - Choix du code.

A la suite de cette étude, on propose deux solutions pour le problème de recherche en tables. L'une se basant sur les travaux respectifs de MM. SIMON

et GUIHO, et l'autre représente une méthode récente reposant sur des statistiques probabilistes, mais qui présente l'inconvénient de ne pas être parfaitement mise au point.

SOUS-PROGRAMME SPI

1 LES CONTRAINTES DU PROBLEME

1.1 - INTRODUCTION

Comme on venait de le voir dans le chapitre "Etude Préliminaire", les deux méthodes vont contribuer pour arriver à la solution cherchée.

La première fonction de SPI est de séparer les syllabes d'un nom conformément à leur valeur phonétique.

Un certain nombre de contraintes entrent en jeu lorsque dans un nom ou un mot français, on désire séparer les différentes syllabes.

1.2 - REGLES

Une syllabe idéale est celle qui se présente sous la même forme du point de vue constitution, partout à l'intérieur d'un nom ou d'un mot, ce qui n'est pas le cas en littérature.

On ne distingue pas entre un S et un SS, entre un PH et un F.

REGLE 1

Une consonne double est équivalente à une consonne simple.

La syllabe la plus simple est celle qui est constituée par une consonne suivie d'une voyelle.

REGLE 2

Dans une succession alternée de consonnes et de voyelles, une coupure avant la consonne permet de séparer les syllabes.

EXEMPLE

CATALOGUE → CA TA LO GUE

MATINEE → MA TI NEE

A un niveau plus compliqué, on a une répartition de lettres qui semble aléatoire à première vue. Or, une étude plus approfondie montre l'absurdité de cette hypothèse.

Deux cas peuvent se présenter :

a - Cas des voyelles

La présence de deux ou plusieurs voyelles les unes à la suite des autres nous amène à introduire la règle suivante :

REGLE 3

Aucune coupure n'est effectuée entre les voyelles, sauf pour le l.

Le cas du l suivi d'une voyelle ou d'un L est assez particulier, il recevra un traitement spécial.

b - Cas des consonnes

Dans une suite de plusieurs consonnes, la règle ne peut être établie qu'après consultation des tables. La coupure pouvant s'effectuer soit avant, soit entre les consonnes. On adopte la règle 4 :

REGLE 4

Dans une suite de consonnes constituant un groupe, la coupure s'effectue entre les consonnes, sauf, si le groupe constitue un phonème établi à l'avance.

EXEMPLE

Pour illustrer les deux cas précédents, prenons les exemples de découpage suivants :

DERANLOT → DE RAN LOT

Il y a une séparation entre le N et le L.

BERTRAND → BER TRAN D

La suite RTR nous donne une coupure entre R et T, et non pas entre le T et le R, car le groupement TR existe comme phonème, de même pour N - D.

On est donc amené à établir une méthode pratique permettant de respecter les règles annoncées plus haut, et de réaliser un découpage logique.

2

CHOIX DE L'ORGANIGRAMME

Pour traiter ce problème, plusieurs solutions sont possibles, des automates présentent un aspect intéressant qui a été réalisé par la thèse de M. COULON (référence 7) en enseignement programmé. Il existe une autre méthode pour localiser les consonnes et les voyelles et pour effectuer le découpage après consultation des tables d'exceptions.

Or, nous avons opté dans cette étude pour une troisième solution qui consiste à réunir dans des tables de structures identiques, l'ensemble de l'analyse syllabo-phonétique, et puis, de procéder à un balayage systématique du nom en se reportant simultanément aux tables, pour puiser les renseignements nécessaires. Une technique de chaînage assez élaborée permet d'utiliser SP1 comme un programme réentrant, jusqu'à épuisement des caractères du nom, indépendamment de sa taille et de l'ordre des caractères.

Il est indispensable avant d'exposer la façon dont SP1 réalise le découpage du nom en syllabes phonétiques, de dire un mot sur la structure des tables.

Question qui sera traitée plus particulièrement au chapitre suivant.

Notons au passage que le constituant, élémentaire de la table, est le double-mot qui est divisé en quatre zones qui sont représentées symboliquement par :

Zone A : représente un caractère

Zone B : représente un code

Zone C, D : représentent deux pointeurs d'aiguillage.

A	B	C	D
---	---	---	---

La première étape consiste en un balayage systématique du nom, caractère par caractère, avec identification simultanée du caractère ou du groupe de caractères, représentés par les tables (comparaison avec le contenu de la zone A).

Lorsqu'il y a égalité, le contenu de la zone B nous renseigne sur le découpage, alors que les zones C et D jouent le rôle d'aiguilleurs nécessaires pour assurer le déroulement du programme.

On rappelle que les tables ont une structure identique, ce qui a permis

l'utilisation d'un programme réentrant, travaillant en boucle fermée, tel qu'il est représenté par l'organigramme suivant.

La comparaison se fait sur un, deux ou plusieurs caractères, suivant la nature de ces caractères, la décision est laissée aux tables.

L'organigramme ainsi présenté ne représente que le principe de la méthode, il est loin d'être complet. Son but c'est de faire comprendre le fonctionnement de SP1 et non pas la construction exacte (voir planche 3).

4

ELIMINATION DES E MUETS

On sait très bien que pour un traitement phonétique d'un nom, la présence des E muets n'est pas vraiment indispensable. Cette présence alourdit énormément le nom. Son élimination nous permet de réduire la longueur du nom d'une syllabe et de s'approcher de la réalité phonétique.

On s'est limité à l'étude de deux cas d'un intérêt pratique, bien que le nombre des cas où l'on peut détecter la présence d'un E muet soit beaucoup plus grand (référence 3).

La raison majeure étant de garder au programme son caractère fonctionnel, sans trop l'alourdir par des règles grammaticales.

E muet final

Si le E muet est précédé d'une voyelle, il sera traité dans le cadre de SP2, en tant qu'appartenant à un groupe vocalique. Si il est précédé d'une consonne, on distingue alors deux cas :

- 1°/ La lettre précédant le groupe consonne - E muet, est une autre consonne.

EXEMPLE

BER THE, DE GRAN GE, LA TOUR TE.

Dans ces conditions, on ne peut rien faire, on garde toutes les syllabes, et le E muet final ne peut être éliminé.

- 2°/ Le caractère précédant le groupe (consonne - E muet), est une voyelle.

EXEMPLE

BA	LLAI	ZE	BA	LLAIZ
PHI	LI	PE	PHI	LIP

Dans ce cas, on élimine le E muet final, et on rattache la consonne précédente à la syllabe précédente.

E muet intérieur

Trois cas sont à écarter où le E ne peut être muet :

1°/ E en deuxième position n'est jamais muet

DE MON, SE GUIN

2°/ Suivi d'une consonne double

LE BE LLOUR, E MMA NUEL

3°/ Encadré de deux consonnes à l'intérieur de la syllabe

LOI SEL, LE VEL

Mis à part ces trois cas, pour détecter un E-muet, il suffit qu'il soit précédé d'une consonne, et suivi d'un blanc indiquant la fin de la syllabe.

Dans ce cas, si la consonne précédente et la consonne qui suit immédiatement le E, forment un phonème consonnique, on dit alors que le E est muet, et on l'élimine.

EXEMPLES

AN CE LIN → AN SE LIN → AN SLIN

CO QUE RI → CO KE RI → CO KRI

DI DE LIN → DI DLIN

LAU BE LAU FFER → LAU BLAU FFER

REMARQUE 1

Le découpage en syllabes a facilité énormément la détection des E muets, chose qui n'aurait pas été facile si on avait le nom brut.

L'organigramme est très simple à réaliser, voir page suivante.

REMARQUE 2

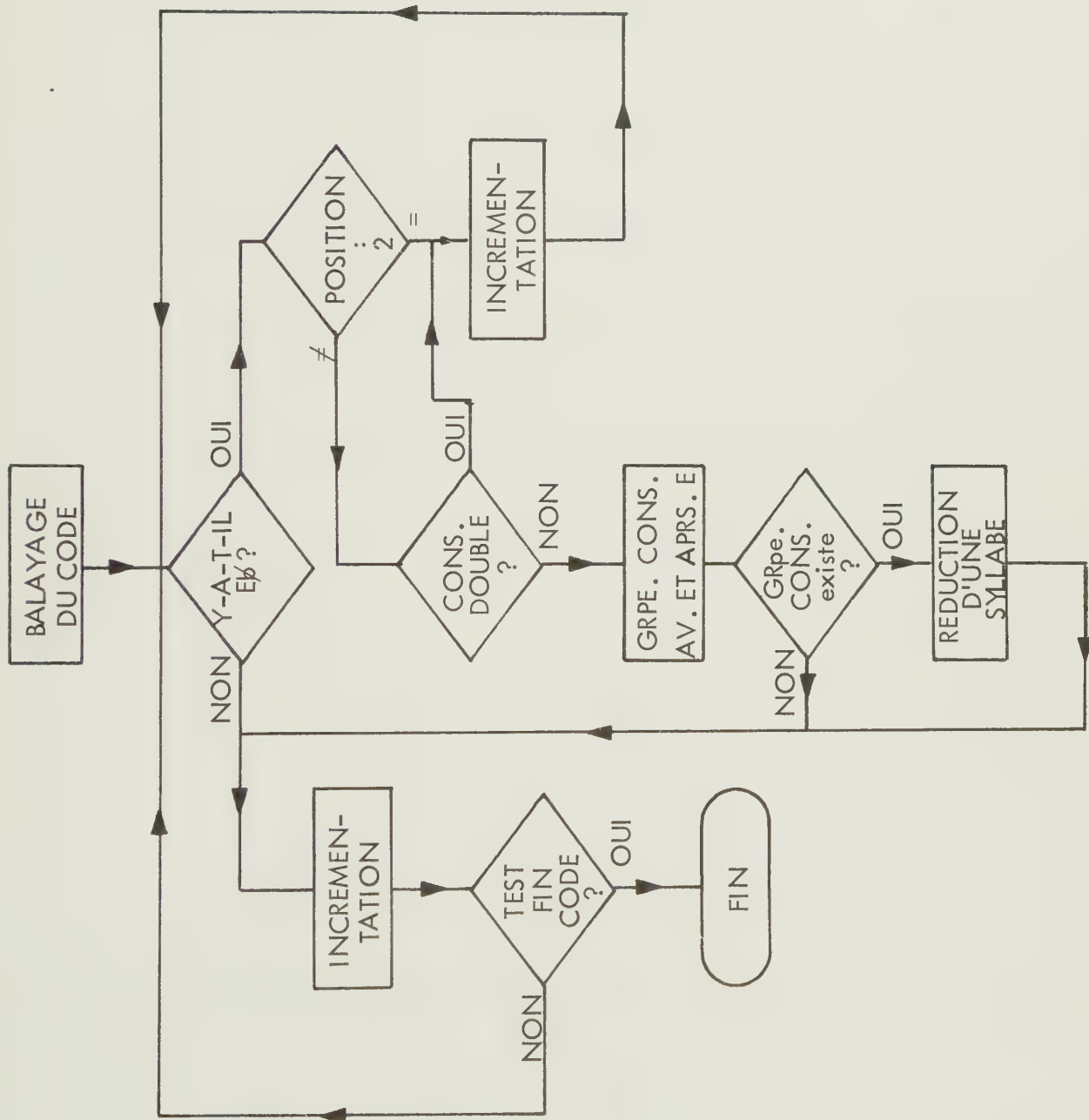
Un deuxième balayage du nom découpé en syllabes est indispensable pour éliminer l'ensemble des E muets dont on gardera trace dans deux compteurs pour la suite.

5

COMPTAGE

Un dernier mot avant d'abandonner SP1, concernant les renvois exprimés sur l'organigramme par ALPHA et BETA.

Il est facile à ce niveau d'inclure une procédure de comptage qui nous permettra de connaître d'une façon précise, la fréquence des phonèmes en tête et à la fin d'une syllabe, comptage qui s'avère très utile pour le choix du code sur lequel on reviendra plus tard.



SOUS-PROGRAMME SP2

1

LES STRUCTURES DES TABLES

Elles sont introduites dans le corps du programme par une procédure externe, elles sont inactives, constituées d'une façon leur permettant de recevoir des nouvelles instructions (des tables ouvertes techniquement). Elles groupent l'ensemble de l'analyse du problème.

1.1 - STRUCTURE

La structure est fonction de l'ordinateur sur lequel le traitement a été fait. Le programme est réalisé en langage d'assembleur IBM 360. Il est donc inévitable de choisir pour éléments de la table des unités en fonction de la structure de la mémoire de l'ordinateur utilisé.

1.1.1 - STRUCTURE INTERNE DE L'IBM 360

L'ordinateur utilisé est du type courant, sauf qu'il a une taille mémoire surabondante (1024 K), on dispose donc de toute la place nécessaire.

La mémoire est adressable octet par octet, la taille de l'octet étant de huit digits binaires et la base de l'adresse est le code HEXA-DECIMAL.

Les octets sont groupés par 2, 4 ou 8 pour former ce qu'on appelle demi-mot, mot ou double-mot respectivement.

L'adressage de base s'effectue de 4096 en 4096 octets, d'où la nécessité de réserver 3 octets minimum pour pouvoir se référer à une adresse.

1.1.2 - STRUCTURE DES TABLES

On a choisi pour élément unitaire des tables de double-mot, taille minimum indispensable, découpé suivant le modèle déjà annoncé dans le cadre de SPI qu'on rappelle ici.



Echelle d'un octet : 

Le premier octet représenté par la zone A, reçoit un caractère alphanumérique. La comparaison du contenu de la zone A avec un caractère du nom nous amène à passer à la consultation des zones B, C, D si égalité il y a. Sinon, on passe à l'élément suivant.

La détection d'un code non-alphabétique en zone A a pour effet :

1°/ Soit de terminer le balayage de la table, et il y a lieu dans ces conditions de passer sur les zones B, C et D.

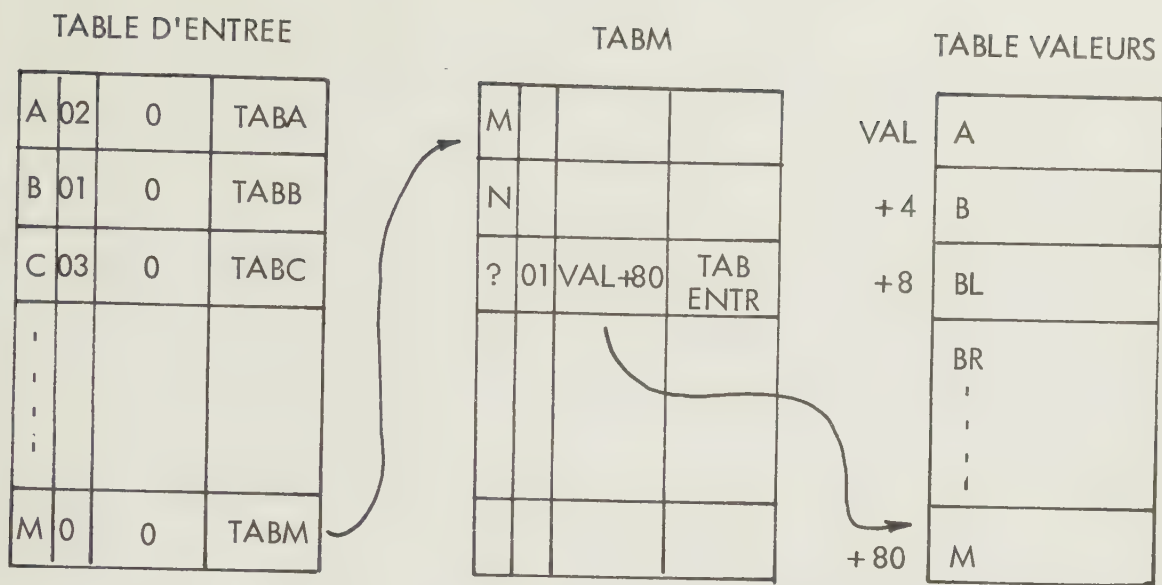
2°/ Soit encore, suivant la valeur du code introduit, d'aller voir si il y a ou non une voyelle qui suit le caractère en consultation.

Le deuxième octet représenté par la zone B, reçoit un code numérique entier à quatre possibilités. Son utilité est de déterminer le point de coupure du nom pour séparer les syllabes.

La zone C, constituée de 3 octets, reçoit soit l'adresse du phonème déjà détecté, soit alors un zéro. Dans ce cas, la valeur du phonème n'est pas achevée, et il y a lieu de continuer le balayage pour détecter un groupement de 2, 3 ou 4 caractères.

Enfin, la zone D reçoit elle aussi, soit une adresse pointant à la tête de la prochaine table à consulter, soit qu'elle reçoit un code indiquant la fin du nom.

Ainsi constituée, une table sera composée d'un ensemble de double-mots convenablement remplis et assemblés séparément.



2

Pour comprendre le fonctionnement des tables, prenons un exemple simple.

Considérons le nom MARTIN.

La première lettre est M, une table d'entrée contenant les 26 lettres de l'alphabet, nous permet après consultation de nous positionner sur les zones en face de M.

La zone B a un zéro, donc pas de coupure.

La zone C a un zéro, donc pas de phonème.

La zone D a un pointeur, donc on se positionne sur la table indiquée.

Après incrémentation, on arrive à la fin de la table où il y a renvoi pour voir si il y a une voyelle.

Quelque soit cette voyelle, il faut qu'il y ait une coupure avant le M (un code dans la zone B). Il y a un phonème (une adresse en zone C), et il y a positionnement sur la table d'entrée (une adresse en zone D), pour continuer le traitement et ainsi de suite.

En plus de la table d'entrée, il nous faut 26 tables de positionnement, et si la décision ne peut se faire au niveau du deuxième caractère, il y a renvoi à des sous-tables de même structure.

LA TECHNIQUE DU CHAINAGE UTILISE EST TRES SOUPLE ET LA TAILLE DES TABLES EST TRES REDUITE, CAR ELLES SONT CONSTITUEES EN FONCTION DES EXCEPTIONS ET NON DES GENERALITES.

EXEMPLE

LA TABLE 'B' EST CONSTITUEE DE 4 ELEMENTS ESSENTIELS : L, R, B, VOY. LA DETECTION D'UN DE CES 4 COMPOSANTS ENTRAINE UNE COUPURE AVANT LE GROUPE AINSI CONSTITUE (BL, BR, BB ou BA, BØ, BU etc..), ET UN RENVOI A UNE VALEUR PHONETIQUE. DANS LES AUTRES CAS, IL Y AURA SEPARATION ENTRE LE 'B' ET LA CONSONNE SUIVANTE.

Les tables contiennent les différentes règles de coupure élaborées au cours de l'étude, notamment les trois règles introduites au cours de l'étude de SP1.

On a utilisé comme support de cette analyse, un fichier réel composé de noms propres choisis dans un annuaire téléphonique où toutes les anomalies des cas ont été profondément étudiées. C'est, peut être, trop dire, car on ne peut jamais grouper tous les cas possibles en se basant sur un fichier de deux à trois mille noms. C'est pour cela qu'on a laissé les tables ouvertes pour permettre l'addition des cas et des règles qui ont pu nous échapper.

On a évoqué le cas du l dans le chapitre précédent, ainsi que le cas de groupements consonniques dont on donnera un aperçu au cours de ce chapitre.

3.1 - TRAITEMENT DU l

On a retenu le l, car il présente des anomalies spéciales suivant sa position à l'intérieur d'un groupe. A savoir :

1°/ Consonne - l - voyelle

La prononciation est analogue à celle d'un groupe composé comme suit :

Consonne l Y VOYELLE

C'est comme si il y avait un Y en plus, en considérant le Y ici comme une consonne et le vrai Y comme un I.

EXEMPLE

LOINTIER → LOIN TI YER

LASABLIERE → LA SA BLI YER

PIANO → PI YA NO

2°/ Voyelle - I - voyelle

Le I disparaît pour donner place à un son équivalent à celui de Y.

EXEMPLE

AIEUL → A YEUL

3°/ Consonne - I - LL

A l'exception de VILLE et MILLE, ce groupement se rencontre souvent dans les noms et le double L sera remplacé par un Y (consonnique).

EXEMPLE

BARILLOT → BA RI YOT

FILLEUL → FI YEUL

4°/ Voyelle - I - LL

Dans ce cas, le groupement ILL en entier est remplacé par le son représenté par Y.

EXEMPLE

BOUILLOT → BOU YOT

VIEILLARD → VI YE YAR D

MAILLOT → MA YOT

Dans les autres cas, le I reste le même. Notons que le Y représente un traitement similaire mais un peu différent.

Les groupements de voyelles sont traités ensembles dans le cadre de SP2.

3.2 - TRAITEMENT DES CONSONNES

1°/ Cas du C et G

Le C et le G, suivis des voyelles E, I, Y, sont considérés comme S et J respectivement, suivis de A ou Ø, c'est un K ou un G.

2°/ Cas du L et R

Ces deux caractères appelés 'liquides' par les linguistes, se trouvent souvent derrière une consonne pour former un groupement phonétique.

EXEMPLE

BL CL DL FL GL

BR CR DR FR GR etc.

3°/ Autres groupements

On a relevé, suivant leur apparition dans la langue française, d'autres groupements à deux ou trois caractères.

EXEMPLE

ST CHM PH SCH TCH SC SP

SPL CHR TH SCHW SM SCHL

STR CHL PHR SCHM SN etc...

A ceci il faut ajouter tous les groupements de consonnes doubles et les groupements vocaliques.

On trouvera la liste en annexe avec le comptage en position principale et terminale de la syllabe.

D'autres lois d'ordre phonétique ont été introduites tel que :

QU → K

H → ELIMINE

On voit très bien que le découpage en syllabes phonétiques rend le nom plus homogène et plus facile à coder. On a donc essayé de dégager des règles générales à partir des particularités, et certaines erreurs volontaires qui ne sont pas des vraies erreurs ont été commises pour l'uniformité de l'étude. Ceci ne représente pas d'inconvénient du moment que l'ensemble du fichier est soumis au même programme.

SOUS-PROGRAMME SP3

1

ELIMINATION DES CONSONNES MUETTES

Au niveau de SP3, le nom a été découpé en syllabes, et les syllabes préalablement codés ce qui ne change en rien la fonction de SP3. Plus loin, nous parlerons du code d'une façon détaillée.

Le décodage réalisé par le sous-programme SP1, isole des consonnes muettes, surtout à la fin du nom et quelquefois à l'intérieur.

La première fonction de SP3 est de détecter ces consonnes isolées et de les éliminer, car elles n'ont aucune valeur phonétique.

EXEMPLES

CESSANT	CE SSAN T ou SE SAN T
GILBERT	JIL BER T
HEROLD	HE ROL D
GRANDCLAUDON	GRAN <u>D</u> CLAU DON
BINDSCHEDLER	BIN <u>D</u> SCHE DLER

Dans ces divers cas, on voit très bien que les consonnes qui s'isolent, du fait du découpage, n'ont aucune importance sur la prononciation du nom, leur élimination n'altère en rien le nom.

2

REDUCTION DES GROUPES VOCALIQUES

La deuxième fonction de SP3 est la réduction des voyelles qui ont été séparées et groupées par SP1 et SP2.

Il s'agit donc de déterminer le nombre de ces voyelles par groupe; on trouve au maximum un groupement de 4 voyelles. Puis, de remplacer ce groupe par son équivalent phonétique, ou un code choisi à l'avance, consigné dans une table de référence.

EXEMPLES

AU	}	représentent le même son
EAU		
ø		

OUI	}	représentent le même son.
UI		

On a relevé une trentaine de groupements possibles pour le français. L'organigramme de SP3 est simple à construire. Il ne représente pas d'intérêt particulier, on le représente schématiquement sur la page suivante.

Toutefois, arrivé à ce stade, le nom se trouve à peu près comme il doit l'être, muni du code équivalent. C'est-à-dire :

Tout groupement de consonnes a été remplacé par un code occupant un octet, ainsi que tout groupement de voyelles. Si on appelle C le code d'une consonne ou groupe consonnique, et V le code d'une voyelle ou groupe vocalique, on peut représenter le nom par une série de C et de V, non uniforme et groupés un peu au hasard. Ceci n'est pas le cas comme il est montré dans l'étude SP4.

SOUS-PROGRAMME SP4

1

ETUDE DE PRINCIPE

On a convenu dans le chapitre précédent, d'appeler par :

V = le groupement d'une ou plusieurs voyelles

C = le groupement d'une ou plusieurs consonnes.

Partant de ces nouvelles définitions, le nom va présenter un certain nombre d'aspects.

EXEMPLES

TAVIAUX

soit

TA VI Y AUX

$C_1V_1 C_2V_2 C_3V_3C_{31}$ avec $V_3 \approx$ AU

ANGENAULT

soit

AN JE NAUL T

VC CV CVC

ECOFFEY

soit

E CO FEI

V CV CV

On distingue donc suivant ce nouveau mode de groupement, quatre sortes de syllabes. A savoir :

- Syllabe commençant par une voyelle + consonne
- Syllabe composée d'une consonne + voyelle
- Syllabe composée d'une voyelle
- Syllabe composée d'une consonne + voyelle + consonne.

REMARQUE

On entend par consonne et voyelle, dans ce qui précède, ce qui a été défini comme C et V.

Comme on l'a vu sur les exemples précédents, le nom se présente sous la forme codée comme une succession de C et de V, non ordonnée. Le but du sous-programme SP4 est de pallier à cette difficulté et de rendre uniforme le code de chaque nom de façon à ce que les C et les V se présentent toujours au même endroit.

Les compteurs introduits dans SP1 nous renseignent sur la nature et la fréquence des C et des V suivant leur position à l'intérieur de la syllabe. On retient notamment le fait que, lorsque la syllabe est composée de C₁ . V . C₂; le deuxième C₂ a une importance faible par rapport au premier C₁. Ce C₂ jouera donc un rôle modificateur sur la valeur de la syllabe.

1.1 - MODIFICATEURS

L'idée de modificateurs est le résultat direct du comptage statistique d'une part, et des règles phonétiques d'autre part. Pour se rendre compte de ce fait, il suffit de faire attention à une personne qui parle au téléphone, par exemple. On arrive facilement à distinguer les consonnes prononcées d'une façon assez forte.

Une analyse approfondie montre que ce sont les consonnes dont on a convenu d'appeler "en position principale" qui ont cette forte sonorité. Les autres viennent ajouter une note de terminaison ou de modification.

L'étude de fréquence de ces caractères vient à l'appui de cette hypothèse. Ceci se comprend facilement lorsqu'on prend deux noms comme MAR TIN et MA TIN qui diffèrent par le R lequel occupe une position de modificateur dans la première syllabe.

Par contre, certains de ces modificateurs ont une importance particulière à cause de leur valeur sonore assez prononcée tel que le S. Cette importance sera mise en valeur par le codage.

Le code est donc transmis à SP4 par SP3, sous forme de syllabes composées de C et de V, séparées par des caractères spéciaux ou blancs.

On sait que la syllabe la plus simple est celle formée d'une consonne suivie d'une voyelle. Soit . CV . L'hypothèse émise ici est la suivante :

- L'absence du C est équivalent à un H aspiré.
- La présence d'un C après un V est **considérée** comme modificateur de la syllabe.
- Le premier C aura le rôle principal dans la syllabe.

L'hypothèse est justifiée par les compteurs introduits lors du découpage.

Nous allons donc passer en revue la réduction des quatre cas de syllabes, sachant que C ou V occupe chacune 1 octet soit 2^8 possibilités de codage.

2.1 - SYLLABE A UN CARACTERE

Ce caractère ne peut être qu'une voyelle ou un groupement vocalique (V), car toutes les consonnes isolées ont été éliminées par SP3.

Dans l'hypothèse émise plus haut, l'absence d'un C précédant un V, va être compensée par l'introduction d'un octet ayant une valeur nulle qui sera appelé C_0 , de façon à ce que la syllabe soit remplacée par une syllabe . C_0V .

REMARQUE

L'isolement d'un V ne peut se produire qu'au début du nom, il est donc traité comme si il était précédé d'un H aspiré.

EXAMPLE

HOGUET HO GUET soit CV CVC

AUGUSTE AU GU ~~STE~~ soit C₀V CVC

2.2 - SYLLABE A DEUX CARACTERES

On distingue deux cas suivant que le premier caractère est un C ou un V,

2.2.1 - CAS .CV.

Dans ce cas il n'y a rien à faire, la forme voulue est déjà acquise, on la garde telle qu'elle est.

2.2.2 - CAS . VC .

Dans ces conditions, on introduit comme dans le cas d'une voyelle isolée, un C_0 nul en tête, de façon à avoir une syllabe C_0VC . , puis le C suivant sa valeur, s'ajoutera au C_0 pour ne former qu'une syllabe $(C_0 + C_1) V$.

Notons au passage que le C peut prendre différentes valeurs, et comme il est en position modificateur, donc son influence sur la syllabe est minime. Sauf pour 2 consonnes (M et N), car précédées d'une voyelle, elles constituent un phonème, et leur présence en C est importante (voir chapitre 'Choix du Code').

2.3 - SYLLABE A TROIS CARACTERES

Ce cas correspond à la succession $C_1 \cdot V \cdot C_2$. C_1 est la valeur principale. V occupe la bonne place. Reste C_2 , qui joue le rôle de modificateur.

La réduction de la syllabe s'effectue de la façon suivante :

$$\cdot C_1VC_2 \cdot \sim \cdot (C_1 + C_{11}) V \cdot$$

Le C_{11} ajouté est un coefficient qui dépend de C_2 . Dans certains cas, ce coefficient pourra s'ajouter à V seulement ou à C_1 et à V.

REMARQUE

L'addition du modificateur sur la valeur principale de la syllabe n'est pas sans intérêt pour notre étude. Car il ne faut pas oublier qu'on cherche à homogéniser les ressemblances entre les noms et non à affiner les différences.

L'organigramme de SP4 nous aide à mieux assimiler cette acrobatie.

Arrivé au niveau de la sortie de SP4, le code apparaît donc comme une succession de consonnes et de voyelles (C et V), uniformément réparties. Chaque groupement de . CV . représente une syllabe, et chaque caractère de la syllabe occupe une position fixée à l'avance. Leur contrôle est donc très facile ainsi que les variations de n'importe quelle syllabe à tous les niveaux sera maîtrisée. Il est donc temps de discuter le choix du code qui rend cette tâche tout à fait automatique.

CHOIX DU CODE

1

CRITERES

Le codage alphabétique tel qu'il est pratiqué, n'est d'aucun intérêt pour notre problème.

Pour coder les différents phonèmes, l'idée nous est venue de faire un rangement phonétique, en se basant sur les valeurs sonores des caractères. Ce qui est plausible à priori, mais il fallait une justification à ce choix.

L'idée de comptage avait un but statistique et phonétique en même temps. On donc lancé le comptage pour un ensemble de 2 300 noms propres au moment de leur découpage en syllabes.

Les résultats obtenus pour les 'C' en position principale sont exacts, alors que les valeurs de 'C' en modificateur ne le sont pas, car on a éliminé par la suite tous les consonnes muettes qui s'isolaient séparément.

Pour les consonnes en position de tête ou en position principale, on a retenu les valeurs suivantes :

L	506	Y	231
R	450		
B	382	P	180
N	375		
M	343		
D	342	T	271
S	307	Z	62
V	215	F	87

A la tête de cette table doit venir le H, car il a une valeur nulle, et est suivi de voyelles qui sont les plus fréquentes de tous.

Or, l'ordre phonétique des groupements qu'on a adopté, est en concordance avec les résultats statistiques. On a introduit en plus les lois de rapprochement de deux consonnes comme le montre le tableau de droite. Par exemple, on a associé à B la lettre P, à D la lettre T, à S la lettre Z, à L la lettre Y (son consonnique) pour des raisons purement phonétiques (voir planche 4).

2

CODAGE

Nous avons convenu de coder chaque groupement représentatif d'un . C . ou d'un . V . par un octet. Or, l'octet est composé de 8 digits binaires. On

dispose donc de $2^8 = 256$ possibilités de codage.

Le choix d'un codage cyclique représente un intérêt particulier pour les mises au point, et la détection d'erreurs par contrôle de parité. En plus, il a l'avantage de faire ressortir la notion de voisinage entre les groupements semblables.

2.1 - CLASSES DE PHONEMES

Dans le tableau précédent, on remarque la tendance des phonèmes à se séparer en 4 grandes classes :

H L Y R qui sont les plus fréquents et les plus rapprochés phonétiquement

M N, B P, D T

V F, J CH

K G, S Z

Ces classes de phonèmes seront représentées par des codes voisins cycliquement, en marquant bien la séparation entre ces classes.

2.2 - SOUS-CLASSES

Chaque caractère à l'intérieur des classes définies plus haut, servira comme base à une sous-classe.

La sous-classe comprendra en plus les éléments composants tel que : B - P, BL, BR, PL, PR etc

On justifiera dans la suite l'hypothèse suivante :

CODER DEUX CARACTERES PROCHES PHONETIQUEMENT AVEC DEUX CODES IDENTIQUES SUR LE DEMI-OCTET DE TETE, ET DIFFERENTS SUR LE DEMI-OCTET DE DROITE OU DE FAIBLE POIDS.

EXEMPLE :

B	<table border="1"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	1	0	1	0	0	0	0	0	= 160
1	0	1	0	0	0	0	0			
P	<table border="1"><tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	1	0	1	0	0	0	0	1	= 161
1	0	1	0	0	0	0	1			

Dans ces conditions la sous-classe aura comme champ de variation tout le demi-octet de droite, ce qui veut dire qu'on impose 16 positions par sous-classe.

La distance entre les sous-classes sera définie par les cases proches cycliquement ou distants de 16 décimal.

Par contre, la distance entre les classes a été choisie en fonction de la place disponible après avoir déterminé les sous-classes de chaque groupe.

REMARQUE

La notion de voisinage entre les différents groupements doit être définie en fonction des modificateurs.

3

MODIFICATEURS

On rappelle que les deux cas possibles où on rencontre des modificateurs dans la syllabe sont :

1°) . VC_2 . qui devient . C_0VC_2

2°) . C_1VC_2 .

La présence de C_2 ayant été préalablement détecté par SP4, on élimine C_2 après consultation d'une table et on ajoutera un certain coefficient sur C_0 , C_1 ou sur V suivant les cas.

On a relevé 5 modificateurs ayant une importance phonétique, à savoir :

L, R, S, M, N.

3.1 - VOISINAGE

On voit donc que les modificateurs peuvent être considérés comme les variables à l'intérieur d'une sous-classe. En se limitant à ces 5 modificateurs + 1 modificateur quelconque, moins important, on voit que pour

coder l'ensemble des cas possibles il nous faut théoriquement :

2^6 positions binaires.

Or, ceci n'est qu'un point de vue théorique, car il suppose tous les cas possibles et imaginables qui puissent se présenter, ce qui n'est pratiquement jamais le cas.

EXEMPLE

BL suivi d'un L en modificateur n'existe pas, mais
BL suivi d'un S est plus fréquent.

Même une étude approfondie d'optimisation ne conduit pas au résultat voulu, car les contraintes du problèmes ne sont pas définies une fois pour tout, et toute rigidité dans les définitions risque de détruire l'intérêt pratique.

En fin de compte, le problème est de trouver la solution pour détecter des ressemblances, à l'aide d'un programme fonctionnel, et non pas de faire une étude théorique qui reste sans application.

REMARQUE

Si le codage du groupe (BL suivi d'un L) est confondue avec celui de BR ou PR, ceci n'est pas grave du moment qu'on reste à l'intérieur de la sous-classe.

Le choix des coefficients à donner aux différents modificateurs doit découler de la solution du système d'équations linéaires. Il faut tenir compte du fait qu'on a réservé 16 positions pour l'ensemble des composants d'une sous-classe.

De toute façon, si on réserve pour les sous-classes des positions de tailles différentes, on déborde notamment sur le demi-octet de gauche et on perd ainsi l'efficacité du traitement.

Cette notion de voisinage est donc très délicate à manipuler. Il faut qu'elle soit optimisée non pas d'une façon théorique mais en fonction des données pratiques.

3.2 - OPTIMISATION

1°) On se limite aux 3 modificateurs, L, R, S et une position pour un modificateur quelconque, autre que ces 3.

2°) Si on prend comme élément de base de la sous-classe le B, qui est nécessairement lié au P à cause de la ressemblance phonétique entre les deux, on voit qu'il faut considérer les cas possibles suivants :

TABLEAU 1

suivi du S, L, R, V en modificateur

$$\begin{array}{cccccccc} \text{B} & " & " & x & x & x & x & " & " & (1) \end{array}$$

$$\begin{array}{cccccccc} \text{P} & " & " & x & x & x & x & " & " & (2) \end{array}$$

$$\begin{array}{cccccccc} \text{BL} & " & " & x & & x & & " & " & (3) \end{array}$$

$$\begin{array}{cccccccc} \text{PL} & " & " & x & & x & & " & " & (4) \end{array}$$

$$\begin{array}{cccccccc} \text{BR} & " & " & x & & & & " & " & (5) \end{array}$$

$$\begin{array}{cccccccc} \text{PR} & " & " & x & & & & " & " & (6) \end{array}$$

Si on appelle :

$$\begin{array}{ll} \text{L} = x_1 & \text{X} = \text{B} \\ \text{R} = x_2 & \text{Y} = \text{P} \\ \text{S} = x_3 & a_0 a_1 \dots a_{15} \\ \text{V} = x_4 & \end{array}$$

Les 16 positions que doivent occuper les composantes de la sous-classe, on voit que les composantes du tableau peuvent se formuler par les équations suivantes :

$$\text{X} = a_0$$

$$\text{Y} = a_1$$

$$\begin{array}{l} (1) \quad \text{X} + x_1 = a_2 \\ \text{X} + x_2 = a_3 \\ \text{X} + x_3 = a_4 \\ \text{X} + x_4 = a_5 \end{array}$$

$$\begin{array}{l} (2) \quad \text{Y} + x_1 = a_6 \\ \text{Y} + x_2 = a_7 \\ \text{Y} + x_3 = a_8 \\ \text{Y} + x_4 = a_9 \end{array}$$

$$\begin{array}{ll}
 (3) & \begin{array}{l} X + x_1 + x_3 = a_{10} \\ X + x_1 + x_2 = a_{11} \\ X + x_2 + x_4 = a_{12} \\ X + 2x_1 = a_{13} \end{array} \\
 (4) & \begin{array}{l} Y + x_1 + x_3 = a'_{14} \\ Y + x_1 + x_2 = a_{15} \\ Y + x_2 + x_4 = a_{16} \\ Y + 2x_1 = a_{17} \dots\dots\dots \text{etc.} \end{array}
 \end{array}$$

L'exploitation pratique de ce système nous amène à éliminer tous les cas qui ne peuvent pas se rencontrer, ou encore les cas ayant une fréquence d'apparition infime.

Si on admet le fait que deux caractères proches phonétiquement doivent avoir le voisinage le plus petit, ceci revient à dire que :

$$(X - Y) = 1$$

Si on réduit les cas d'apparition les plus fréquents marqués en x dans le tableau 1, notre système se réduit à :

$$\begin{array}{ll}
 X = a_0 & \\
 Y = a_1 & \\
 x_1 = a_2 & \\
 x_2 = a_3 & \\
 x_3 = a_4 & x_1 + x_2 = a_{12} \\
 x_4 = a_5 & x_1 + x_3 + 1 = a_{13} \\
 x_1 + 1 = a_6 & x_2 + x_3 + 1 = a_{14} \\
 x_2 + 1 = a_7 & x_1 + x_2 + 1 = a_{15} \\
 x_3 + 1 = a_8 & \\
 x_4 + 1 = a_9 & \\
 x_1 + x_3 = a_{10} & \\
 x_2 + x_3 = a_{11} &
 \end{array}$$

Il s'agit donc de résoudre ce système pour les valeurs entières comprises entre (2, 15), parmi les solutions possibles on retient les valeurs :

$$x_1 = 4$$

$$x_2 = 6$$

$$x_3 = 8$$

$$x_4 = 2$$

3.3 - CODAGE DES MODIFICATEURS

On distingue à la suite de cette étude d'optimisation 3 cas possibles :

1er Cas

La présence d'un modificateur quelconque autre que (L, R, S, M, N) correspond à la solution $x_4 = 2$, ce qui a pour effet l'élimination de la consonne en question, et l'addition de + 2 sur l'octet représentatif de la valeur principale.

2ème Cas L, R, S

L et R correspondent aux solutions $x_1 = 4$ et $x_2 = 6$, on procède de même que dans le premier cas. Pour coder un B suivi d'un L, par exemple, on ajoutera + 4 sur la valeur donnée à B, et + 1 sur la voyelle qui se trouve entre B et L, alors que pour BL, on ajoutera + 4 sur la valeur de B tout simplement.

Ce procédé nous permet de détecter une ressemblance assez poussée entre :

BLA et BAL

BR et BER par exemple.

REMARQUE 1

Cette façon de coder nous permet de résoudre le problème d'inversion de certains caractères tel que le L et le R.

Le S ne pose pas de problèmes particuliers à cause de sa sonorité assez forte. On lui a accordé la solution $x_3 = 8$ et on procède comme précédemment.

REMARQUE 2

Le tableau de la page suivante illustre cette méthode de codage. On n'a pas voulu le surcharger avec tous les cas possibles, on note au passage que les lignes représentent les sous-classes définies plus haut.

REMARQUE 3

Du moment où tous ces modificateurs viennent s'ajouter au code principal, ils vont donc modifier le demi-octet de droite et la ressemblance n'est pas altérée.



3ème Cas M, N

Le cas du M et du N ne peut être traité comme les précédents, car en modificateurs, M ou N se trouvent précédés d'une voyelle, et l'ensemble forme un phonème.

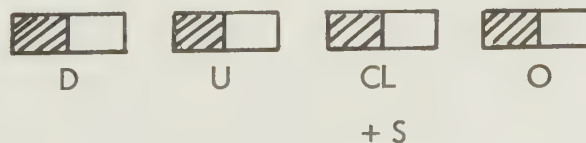
Or, on sait très bien que c'est le N ou le M qui ont plus d'importance par rapport à ce qui précède, d'où l'idée de coder les voyelles en se limitant à l'intérieur de l'octet des 7 digits de droite, et d'ajouter un 1 en tête de l'octet, soit l'équivalent de 80 hexadécimal pour un éventuel caractère nasal.

Explications pour mieux comprendre la notion de ressemblance introduite jusqu'ici, en prenant un exemple courant :

EXEMPLE

DUCLOS → DU CLOS

en représentation schématique il est étendu sur 4 octets



On voit nettement que ces ressemblances en tête d'octets permettent de retrouver facilement un nom s'il y a eu inversement d'une lettre ou même omission. Ainsi le nom DUCLOS est rapproché avec tous les noms qui

peuvent se constituer à partir des tables suivantes :

							
T	+ R	U		K		AU	
D		OU		CR		OT	
DL	+ V			QU		AUX	
DR				CL ...		AULT ...	etc.

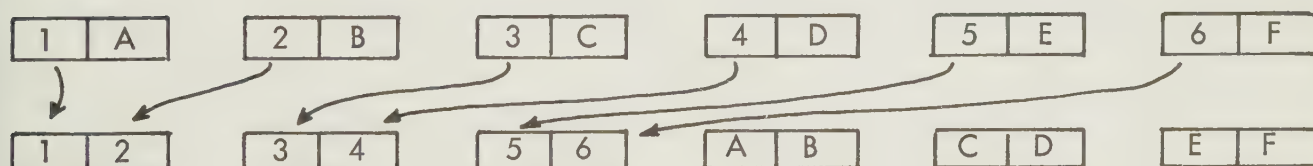
On est loin d'avoir énuméré tous les cas possibles. De toute façon, une étude systématique peut être faite sur les erreurs écrites ou phonétiques. La méthode étudiée offre une solution adoptable à n'importe quel problème de ce genre avec des légères modifications. On voit très bien qu'à ce niveau là, du moment où on connaît les positions exactes de nos consonnes et voyelles, et avec un codage adapté, on peut maîtriser les variations et même les altérations ou omissions de n'importe quel nom français (sauf pour des noms d'origine étrangère).

Pour des noms d'origine latine, autres que le français, la méthode reste valable, avec modification des tables en fonction des règles phonétiques et syllabiques de chaque langue.

Le but ayant été atteint, l'exploitation du code en question dépend de la nature de l'application. On a réalisé un programme (SP5) assez compliqué, dont on ne fait que citer les fonctions qu'il réalise pour illustrer un moyen d'exploitation.

On dispose donc d'un nom propre et de son code traité par l'ensemble des sous-programmes précédentes. On sait que s'il y a ressemblance entre ce nom et un autre nom, cette ressemblance se trouve localisée dans les demi-octets gauches des codes respectifs de chaque nom.

La première fonction de SP5 est donc d'extraire les demi-octets de gauche du code, et de les ranger les uns à côté des autres jusqu'à épuisement, puis de faire de même avec les demi-octets de droite.



De cette façon on a les ressemblances en tête du code. On peut même se fixer une largeur d'un double-mot pour le code, et cadrer ce rangement d'octets soit à gauche soit à droite.

On voit donc que en partant de la droite et en éliminant demi-octet par demi-octet, on peut décélérer tous les noms ressemblants sur 4, 3, 2 ou 1 syllabes.

Une deuxième fonction réalisée par ce sous-programme est comme suit :

Ranger les différents octets, digit par digit, en commençant par le digit de tête de chaque octet jusqu'à épuisement des octets. Ceci a pour but de décélérer des variations infimes entre les noms en se basant sur les variations des digits de plus faible poids.

Une troisième façon de réarrangement est celle qui permet une lacheté un peu plus accentuée. On réarrange les octets par couple de 2 digits.

La première fonction est de loin la meilleure parce qu'elle isole partie ressemblance dans un mot et partie différences dans un autre mot contigu ceci permet une utilisation assez souple suivant les besoins. Sur la planche 7, on a fait varier le code résultant demi-octet par demi-octet, en partant de la droite et on a consigné les noms semblables dans des colonnes séparées par des espaces blancs.

RECHERCHE EN TABLE
PAR LES FONCTIONS G

1 RAPPEL

L'application proposée dans ces pages repose sur les travaux respectifs de Monsieur le Professeur J.C. SIMON et de Monsieur GUIHO (référence 5).

L'idée est la suivante :

A partir d'une courbe de rangement en mémoire représentative de l'adresse relative en fonction de la valeur de l'enregistrement. A l'aide d'approximations on cherche une fonction qui approche cette courbe de rangement en mémoire.

La fonction est connue par un certain échantillonnage, et l'idée est d'habiller chaque point d'échantillonnage par une petite fonction à support compact local. . De sorte que pour avoir la valeur en un point donné, il n'y a qu'à faire intervenir un certain nombre de points d'échantillonnage voisins. Ces petits fonctions à support compact seront nommés "fonction G". (Pour plus de détails sur les fonctions G, voir thèse GUIHO en Mathématiques appliquées, Avril 1970) (Références 5 et 11).

Pour l'application en question, nous sommes partis non pas du nom lui-même mais de son code déjà établi par le programme précédent. Ce code a été pondéré pour pouvoir réaliser une variation numérique ordonnée. La base de numération prise était la base hexadécimale et non une base alphabétique (26).

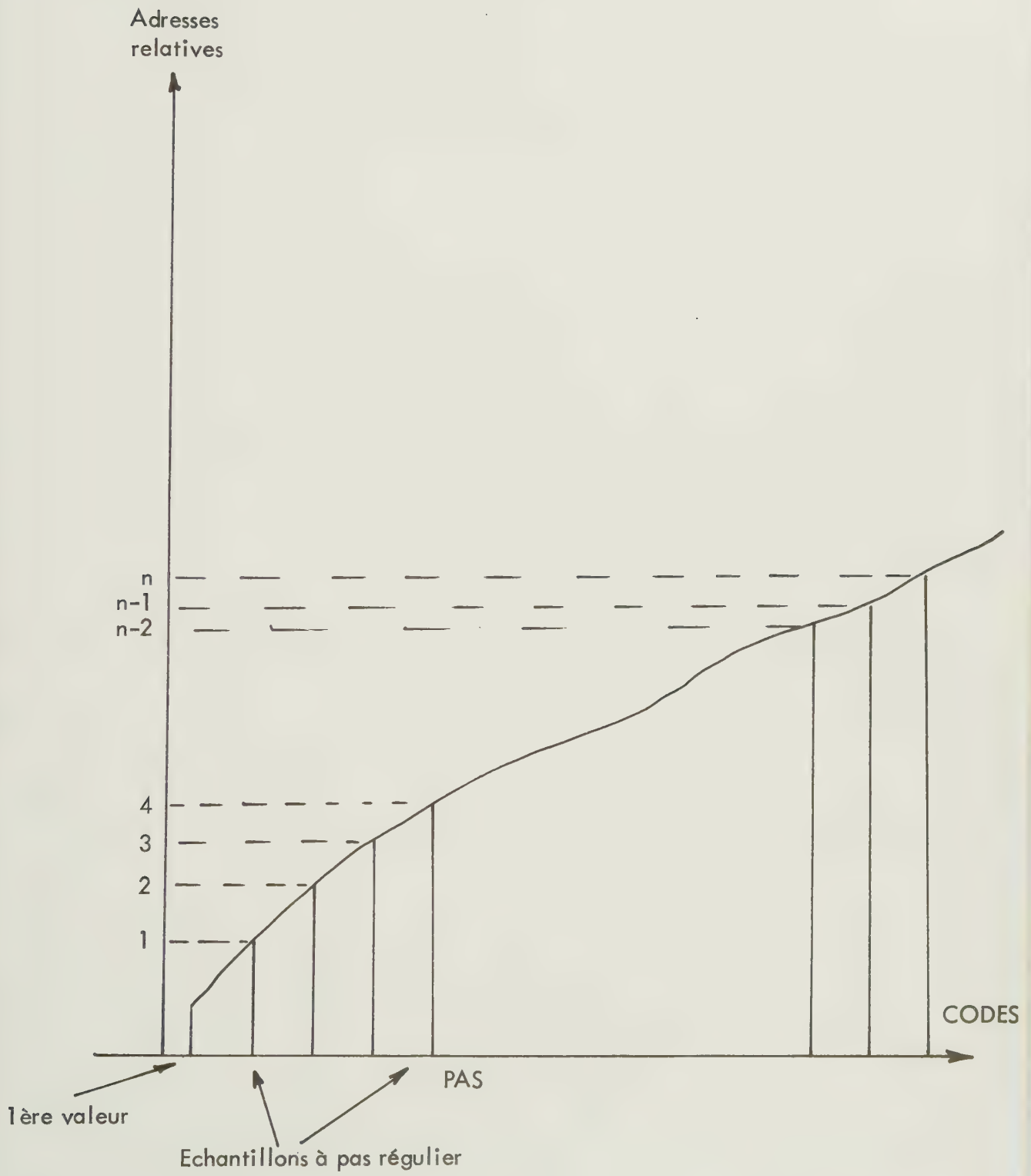
Il est donc indispensable d'avoir un rangement régulier, croissant par exemple, des différents codes, chaque code avec le nom propre correspondant occupera une position mémoire.

Pour le fonctionnement de l'échantillonnage à pas régulier, il est indispensable de connaître l'adresse du premier enregistrement et celui du dernier enregistrement. (figure 1).

L'adresse des échantillons est déterminée par la fonction G3. On a effectué plusieurs applications. La précision est d'autant meilleure que le nombre des échantillons est plus grand; par contre; elle est indépendante de la valeur directe du code. Un bon résultat est obtenu pour ces fichiers avec un nombre d'échantillons égal au quart de la taille du fichier exploité.

Différentes observations ont pu être relevé au cours de cette étude.

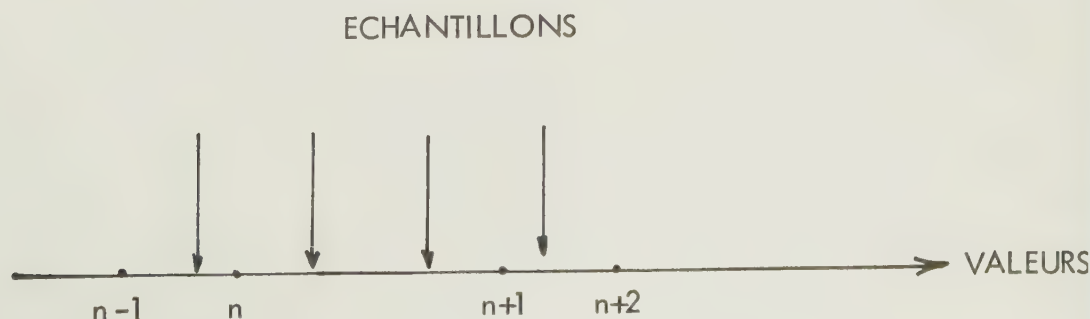
Figure 1



- 1°) La taille de l'échantillonnage étant fonction de la taille du fichier, il est normal qu'elles croissent ensemble. On reste toutefois, dans les limites d'encombrement permises par le "HASH CODING". Il y a ici un avantage certain sur la dichotomie globale, appliquée sur un fichier d'une certaine taille.
- 2°) Il serait souhaitable de trouver un pas d'échantillonnage qui, sans présenter trop de points de collision, puisse couvrir l'étendue du fichier avec des bons résultats. De toute façon, les collisions entre échantillons ne gênent pas, du moment où ils occupent des positions théoriques. Toutefois, on a rencontré des difficultés aux bords qui peuvent être résolus par des fonctions G dissymétriques.

On était donc obligé à modifier le sous-programme pour admettre les collisions entre les échantillons pour conserver la notion de la régularité de l'échantillonnage.

Il aurait fallu un échantillonnage à pas irrégulier, mieux adapté à ce problème.



3°) En outre, on a supposé que les différentes valeurs sont croissantes et occupent des positions successives de la mémoire. Ce qui n'est pas le cas, car le but du traitement des noms propres par le programme précédent était de créer des codes permettant de déceler les ressemblances entre les noms. De ce fait, beaucoup de noms se trouvaient munis de codes identiques ou très peu différents les uns des autres, de tel façon que la courbe de la figure 1 n'est plus optimum et présente des irrégularités à tangence infini (voir figure 2).

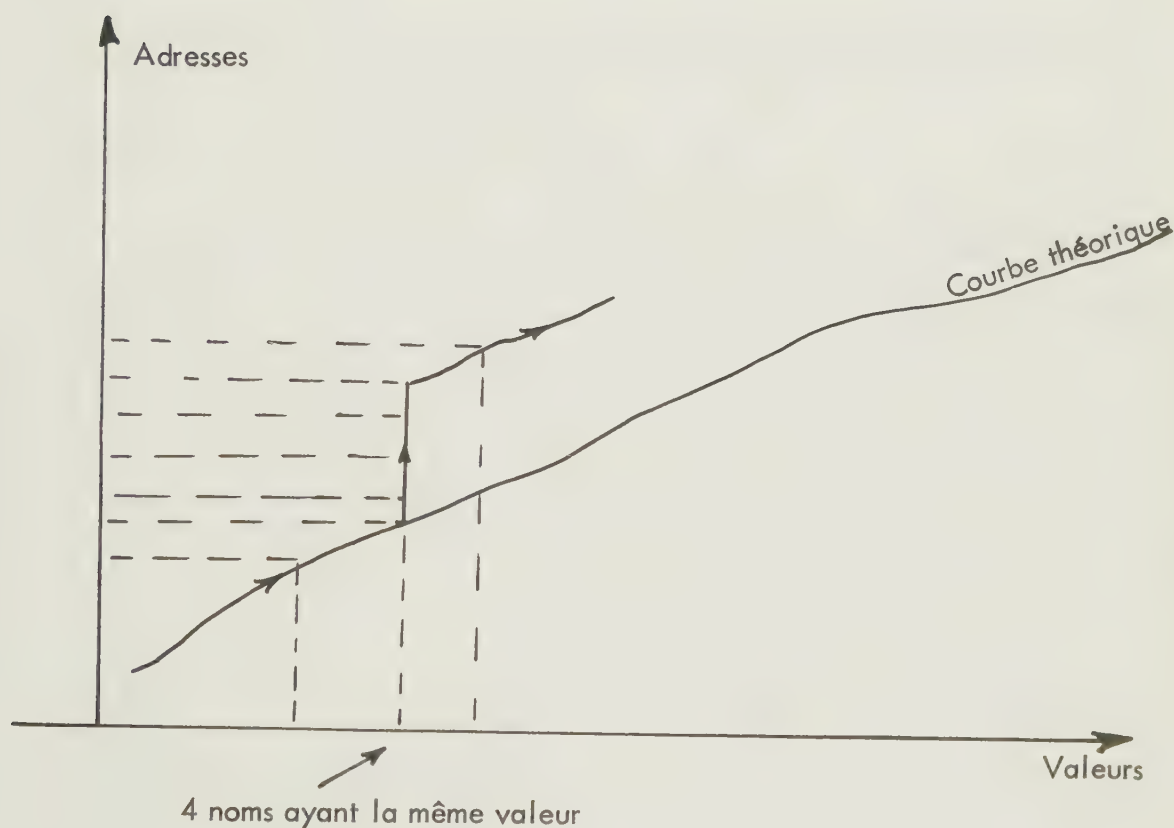


Figure 2

Dans ces conditions, l'approximation par les fonctions G se trouve moins bonne. Une solution intéressante consiste à introduire des trous dans la table.

4

SOLUTION PROPOSEE

Une fois le fichier créé et ordonné, pour exploiter la recherche en table par les fonctions G, on était amené à scinder ce fichier en deux.

Un sous-programme qu'on appelle SPG, réalisera cette distinction en se basant sur la notion des listes enchaînées.

Une première lecture des codes nous amène à renvoyer en FICH2, tous les codes (et noms) se ressemblant à 4 octets près, avec établissement d'un pointeur dans le fichier 1 avec le nom auquel il ressemble dans fichier 2.

C'est la notion de chaînage par adresses par l'intermédiaire de cellules uniforme.

CELLULE

AD1	NOM	AD2
-----	-----	-----

AD1 = adresse du nom

AD2 = adresse de chaînage

Lorsqu'on rencontre deux ou plusieurs noms ressemblants dans FICH1, on conserve un seul, et on renvoie les autres dans FICH2.



L'enchaînement dans FICH2 entre noms ressemblants s'effectue de la même manière.

Les essais pratiques nous fournissent le résultat suivant : sur 2 300 noms on recueille 1 670 en FICH1 et le reste en FICH2.

La recherche en table appliquée sur FICH1 avec 400 échantillons à pas régulier, on a trouvé des résultats avec une meilleure moyenne en adresse relative.

$$M = \frac{\sum |A - C|}{\sum \text{NOMS}} = \frac{\sum |A - C|}{1670} = 4,56$$

avec A : adresse vraie

C : adresse calculée

Planche 6 : adresses vraies comparées aux adresses calculées

Planche 7 : illustration CHAINAGE

ETUDE PROBABILISTE

1

INTRODUCTION

Il a été exposé au cours de l'étude précédente qu'il est indispensable d'avoir un fichier ordonné pour la recherche en table. On conservera cette notion d'ordre d'une importance fondamentale pour la suite et on se propose de démontrer une méthode originale et rapide qui nous permet d'avoir un accès direct à un enregistrement quelconque.

On a remarqué au cours de l'étude fondamentale que l'ordre des lettres ou de groupes de deux lettres n'est pas tout à fait aléatoire. Certains groupements sont plus fréquents que d'autres, ainsi que la succession des lettres à l'intérieur d'un mot semble suivre des lois dont on ignore l'origine.

On se demande donc s'il existe un moyen qui nous permet de localiser un enregistrement, en tenant compte :

- de l'arrangement des lettres constituant l'enregistrement
- de la fréquence des groupements conditionnels, des caractères pris deux à deux.

C'est donc chercher une loi de variation des adresses d'enregistrements qui soit conforme avec le rangement en question.

2

PRINCIPE

Dans une succession de caractères alphabétiques, on peut grouper les caractères deux à deux de deux façons sans détruire la notion de suite, à savoir :

- groupements indépendants
- groupements liés.

EXEMPLE : MARTIN

Selon le premier mode de groupement on a 3 groupes :

MA - RT - IN

et selon le deuxième mode de groupement on a 6 groupes :

ØM - MA - AR - RT - TI - IN

Ce dernier mode de groupement exprime que l'apparition de chaque caractère est conditionnée par la présence du caractère précédent. Cette idée de groupes conditionnels peut s'exprimer comme suit pour l'ensemble de l'alphabet :

ETANT DONNE UN CARACTERE DE L'ALPHABET, QU'ELLE EST LA PROBABILITE POUR TROUVER UN DEUXIEME CARACTERE QUELCONQUE DE CET ALPHABET QUI LUI SUCCEDE DANS LA SUITE CONSTITUANT LE NOM OU L'ENREGISTREMENT.

EXEMPLE

Si on définit l'alphabet comme un ensemble $\{ \text{A B C} \dots \text{Z} \}$ de vingt-sept caractères (le caractère blanc étant inclus dans cet ensemble), et si AR est un groupe conditionnel de la suite des caractères MARTIN, on peut dire que la probabilité pour trouver un caractère quelconque qui suit A est de :

$$\begin{cases} p = 1 & \text{pour } R \\ p = 0 & \text{pour } \{ \text{ALPHABET} \} - \{ R \} \end{cases}$$

ce qui est représenté conventionnellement par :

$$p(R | A) = 1$$

Il est donc souhaitable de pouvoir exploiter ces résultats non pas pour un enregistrement donné mais pour l'ensemble du fichier et de pouvoir calculer la probabilité de présence d'un couple à l'intérieur de l'ensemble en question.

Si on admet l'hypothèse suivante qu'on justifiera plus loin :

LA PROBABILITE DE PRESENCE D'UN COUPLE DE CARACTERES EST INDEPENDANTE DE SA POSITION A L'INTERIEUR DE L'ARRANGEMENT OU DU NOM SAUF POUR LES DEUX PREMIERS CARACTERES.

alors on pourra dire qu'il suffit pour résoudre le problème évoqué plus haut, de faire le cumul des probabilités des différents groupements au fur et à mesure qu'ils se présentent. Il est donc inutile d'avoir un fichier trié ou arrangé à l'avance.

Une table ou une matrice ayant une dimension de 27×27 , nous permet de recueillir les résultats statistiques de l'étude.

2.1 - MATRICE 27×27

La matrice en question est destinée à recevoir les probabilités cumulées des différents groupements; la raison pour laquelle on a pris une matrice 27×27 est justifiée par le rôle important que joue le caractère blanc devant le nom pour un positionnement alphabétique.

On représentera successivement en ligne le cumul des probabilités conditionnelles de tous les caractères de l'alphabet, lorsque le premier est connu pour un fichier donné.

Ainsi convenu, un groupe $B\emptyset$ rencontré, aura pour effet l'addition d'un 1 dans la case $\alpha(b, \emptyset)$, où :

α = élément de la matrice correspondant à $p(O|B)$

b = la ligne correspondant à B

ϕ = la colonne correspondant à $\emptyset$

soit schématiquement :

1er Caractère	2ème Caractère													
	∅	A	B	C	D	E	F	G	H		O		Z	
∅														
A				$p(C A)$			$p(F A)$							
B													1	
C														
D			$p(B D)$									$p(O D)$		
.														
.														
.														
O							$p(F O)$							
.														
.														
.														
Z														

$$\alpha(b, o) = \alpha(3, 16)$$

ligne , colonne

Pour que cette matrice ait un sens de probabilités, il suffit de diviser chaque élément d'une ligne par la somme de tous les éléments de cette ligne. On obtient ainsi la probabilité conditionnelle pour trouver un groupe de deux lettres à l'intérieur du fichier (voir planche 8).

2.2 - EXPLOITATION DE LA MATRICE

Cette matrice est utilisable d'une façon efficace dans la mesure où elle permet un calcul d'adresses permettant un positionnement direct sur l'enregistrement.

Suivons sur la planche 8 l'évolution des probabilités conditionnelles pour le caractère D. On remarque qu'aucune de ces probabilités n'est supérieure à un. Et mieux encore, la somme de tous les éléments d'une même ligne est égal à 1, soit pour D :

$$\boxed{\sum_{\alpha=A}^Z p(\alpha | D) = 1} \quad (1)$$

Formule qui exprime que l'ensemble des noms commençant par D, quels que soient ces noms, ils forment un groupement indépendant dans lequel les différents groupes de 2 caractères ont des chances d'apparition plus ou moins variées, mais dont la somme est égale à 1.

Le rôle du caractère blanc au début du nom permet de connaître la probabilité du groupe des noms commençant par D par rapport à l'ensemble du fichier.

On a vu que la probabilité de présence d'un groupe de 2 caractères est fonction du premier caractère du groupe et de la somme de tous les probabilités composantes donc de (1).

En d'autres termes, cette probabilité conditionnelle du groupe en question est une fonction de la somme représentée par (1). Si donc on convient d'assimiler cette somme à un chemin ou interval, INT, la probabilité $p(i | j)$ représentera un pourcentage de cet interval.

EXEMPLE

Si on considère la suite MARTIN et si, dans la matrice l'élément correspondant à :

$$p(M | \emptyset) = 0,12 \quad (2)$$

ceci voudra dire que les noms commençant par M , occupent 12 % de l'encombrement du fichier.

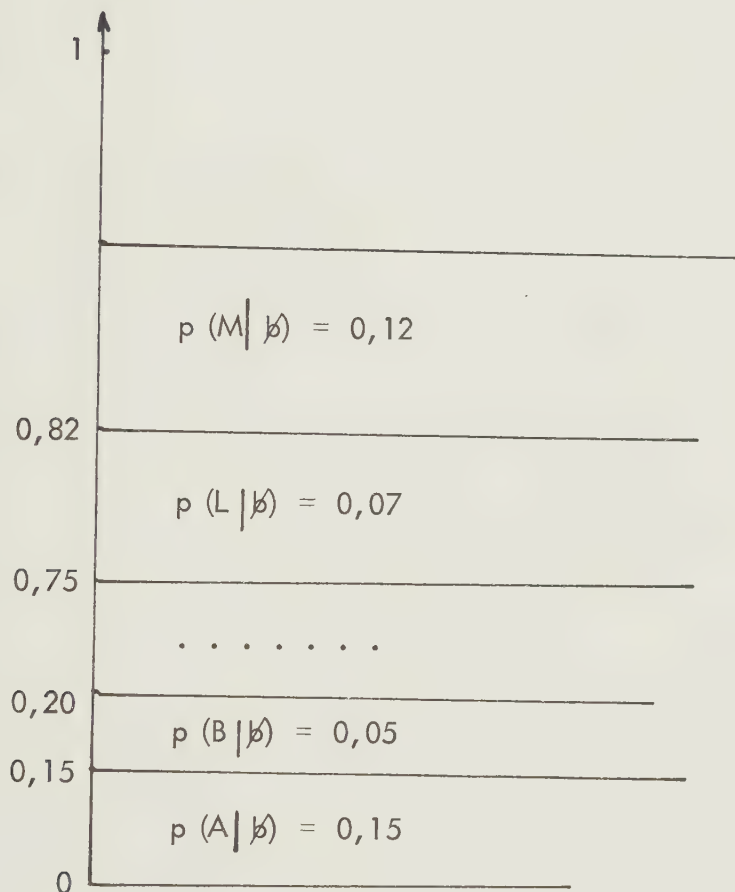
Mais pour localiser le début des noms commençant par M , il est indispensable de connaître la somme des probabilités de tous les noms commençant par :

$\text{A, B, C} \dots \text{jusqu'à L.}$

soit :

$$p(A | \emptyset) + p(B | \emptyset) + p(C | \emptyset) + \dots + p(L | \emptyset) = \sum_{i=A}^L p(i | \emptyset) \quad (3)$$

Donc, les noms qui commencent par D et qui représentent un pourcentage de 12 % (2), doivent être précédés par la somme (3), donc ils ont une position bien définie à l'avance.



Un premier positionnement est réalisé par ce premier groupement pour connaître le positionnement des noms commençant par MA, le nouveau interval de variation sera :

$$\boxed{p(M | \emptyset) = 0,12} \quad (4)$$

et la $p(A \mid M)$ représentera une fraction de (4) dont le positionnement dépend de ce qui précède.

Il y a ici une idée de récurrence entre les probabilités des différents groupements qui jouent le rôle de facteurs numériques fractionnaires d'occupation mémoire.

L'exploitation de la récurrence entre les probabilités de groupements constitutifs du nom est facilitée par l'utilisation d'une matrice substituée de la première (voir planche 9) où un élément d'une ligne est égale à la somme de tous les éléments précédents de cette ligne.

Ainsi un élément α_{ij} de la première matrice devient

$$\alpha_{ij} = \sum_{N=1}^i \alpha_{iN}$$

Si le groupe en cours de traitement est celui de rang n , et si la probabilité est désignée par l'élément de la matrice $E(L, K)$ avec :

L : rang de la ligne

K : rang de la colonne,

la récurrence introduite précédemment peut s'exprimer par l'algorithme suivant :

$$\begin{aligned} A_n &= A_{n-1} + E_n(L, K-1) \times I_{n-1} \\ I_n &= I_{n-1} \times (E_n(L, K) - E_n(L, K-1)) \end{aligned}$$

I_n = interval

Les E_n sont les éléments de la matrice représentée par la planche 9.

A_n = adresse partielle, initialisée à zéro au début de chaque nom, soit :

$$A_0 = 0 \text{ pour } n = 1$$

$E_n(L, K-1)$ = est la somme des probabilités conditionnelles précédant le groupe en cours de traitement

I_n = c'est l'intervalle de variation de l'ensemble de noms ayant le groupement en question.

$E_n(L, K) - E_n(L, K-1)$ = probabilité du groupement en cours de traitement.

Précisons la notion d'intervalle introduite précédemment sur un exemple :

DUPONT

$I_0 = 1$, c'est la taille du fichier rapportée à l'unité

A_1 = le début des noms propres commençant par D

I_1 donne la longueur de l'intervalle sur lequel s'étendent les noms commençant par D.

A_2 donne le début des noms commençant par DU à l'intérieur de I_1 .

I_2 donne la longueur de l'intervalle sur lequel s'étendent les noms commençant par DU

et ainsi de suite jusqu'à épuisement des caractères.

	1°	2°	3°	4°	5°	VALEUR ADOPTÉE	VALEUR MOYENNE
AN	2400	1178	3171	1486	1842	1717	2015
BE	2548	2500	3289	3333	1935	2834	2721
CH	2979	8889	3448	4675	5143	4241	5026
CO	2979	0279	1839	0649	0000	3147	1386
DE	3363	2000	1000	2830	2353	3025	2307
FO	2769	0000	1200	1304	2000	1844	1818
GA	2206	0000	1236	0845	1111	1366	1374
GU	3162	5000	1910	0141	0833	2061	2209
HA	2532	3284	2581	1930	1800	2408	2425
HE	3291	2836	2097	4035	4800	3371	2812
IM	4444	0327	0391	0110	0130	0231	1080
IS	2222	0408	1500	0824	0000	0932	0950
JA	3514	4286	1667	5000	0000	3214	3600
LA	4934	2969	1101	1942	1206	2111	2430
LE	2314	1400	0596	2961	2057	2195	2000
MA	5000	0500	3000	3607	4737	4054	3350
MO	2778	3500	0303	1639	0000	2359	2055
NA	3077	1389	1376	1625	1188	1210	1730
NO	3846	0550	1005	1000	0693	1998	1418
PE	3412	3333	1379	1579	2000	2919	2340
RA	2222	1884	0821	1341	1333	1275	1520
RE	2821	1957	1026	2626	1500	2005	1986
RO	3070	1300	0968	1341	0583	1194	1452
SC	2129	2083	0877	0259	0789	1147	1227
SE	3161	0417	0439	1466	1711	1769	1480
TA	2314	2941	1209	0924	0513	1253	1600
TE	2893	2941	1099	2017	2821	2255	2550
UL	3330	0361	0778	1000	1168	0935	1320
UR	3330	0825	1278	0375	1387	1016	1410
UZ	3330	0155	0389	0000	0000	0160	0774

REMARQUE 2

On a relevé au cours de cette étude même les groupements ayant la fréquence d'apparition la plus élevée, et on a remarqué qu'ils sont très stables et leur probabilité ne varie pas en fonction de la position où ils apparaissent. Ce sont les groupements ayant pour premiers caractères :

A - E - I - L - N - O - R - U

	1°	2°	3°	4°	5°	VALEUR ADOPTÉE	VALEUR MOYENNE
AN	2400	1178	3171	1486	1842	1717	2015
BE	2548	2500	3289	3333	1935	2834	2721
CH	2979	8889	3448	4675	5143	4241	5026
CO	2979	0279	1839	0649	0000	3147	1386
DE	3363	2000	1000	2830	2353	3025	2307
FO	2769	0000	1200	1304	2000	1844	1818
GA	2206	0000	1236	0845	1111	1366	1374
GU	3162	5000	1910	0141	0833	2061	2209
HA	2532	3284	2581	1930	1800	2408	2425
HE	3291	2836	2097	4035	4800	3371	2812
IM	4444	0327	0391	0110	0130	0231	1080
IS	2222	0408	1500	0824	0000	0932	0950
JA	3514	4286	1667	5000	0000	3214	3600
LA	4934	2969	1101	1942	1206	2111	2430
LE	2314	1400	0596	2961	2057	2195	2000
MA	5000	0500	3000	3607	4737	4054	3350
MO	2778	3500	0303	1639	0000	2359	2055
NA	3077	1389	1376	1625	1188	1210	1730
NO	3846	0550	1005	1000	0693	1998	1418
PE	3412	3333	1379	1579	2000	2919	2340
RA	2222	1884	0821	1341	1333	1275	1520
RE	2821	1957	1026	2626	1500	2005	1986
RO	3070	1300	0968	1341	0583	1194	1452
SC	2129	2083	0877	0259	0789	1147	1227
SE	3161	0417	0439	1466	1711	1769	1480
TA	2314	2941	1209	0924	0513	1253	1600
TE	2893	2941	1099	2017	2821	2255	2550
UL	3330	0361	0778	1000	1168	0935	1320
UR	3330	0825	1278	0375	1387	1016	1410
UZ	3330	0155	0389	0000	0000	0160	0774

REMARQUE 2

On a relevé au cours de cette étude même les groupements ayant la fréquence d'apparition la plus élevée, et on a remarqué qu'ils sont très stables et leur probabilité ne varie pas en fonction de la position où ils apparaissent. Ce sont les groupements ayant pour premiers caractères :

A - E - I - L - N - O - R - U

Un adressage qui réduit l'erreur de positionnement pourra se baser sur ces groupements stables en permettant un saut de caractères dans le balayage du nom.

Le traitement phonétique du début pourra avoir des extensions intéressantes dans le cadre de cette étude. Le nom ayant été réduit de façon à mettre en évidence les caractères ayant une importance phonétique. On voit donc que le saut de caractères préconisé pourra se faire en se limitant au demi-octet de gauche ce qui réduit d'une part la taille de la matrice (17 x 17) et permettra d'autre part, un calcul d'adresses par groupes de noms.

Le but recherché était de trouver, s'il existe ou non une matrice de transition permettant de passer d'une position à une autre avec une meilleure précision, mais ceci n'a pas pu être prouvé à cause de la taille réduite de notre fichier, il en fallait au moins 50.000 noms pour voir clair.

D'autre part, au cours du calcul des adresses, on a rencontré des points de collision d'un certain ordre. C'est-à-dire que plusieurs noms rapprochés avaient des adresses identiques. Une étude plus approfondie nous a amené à constater qu'après le cinquième groupe, l'algorithme choisi ne distinguait plus les différences entre les adresses de deux noms semblables dans leur partie de tête, par exemple :

MARTIN

MARTIANO

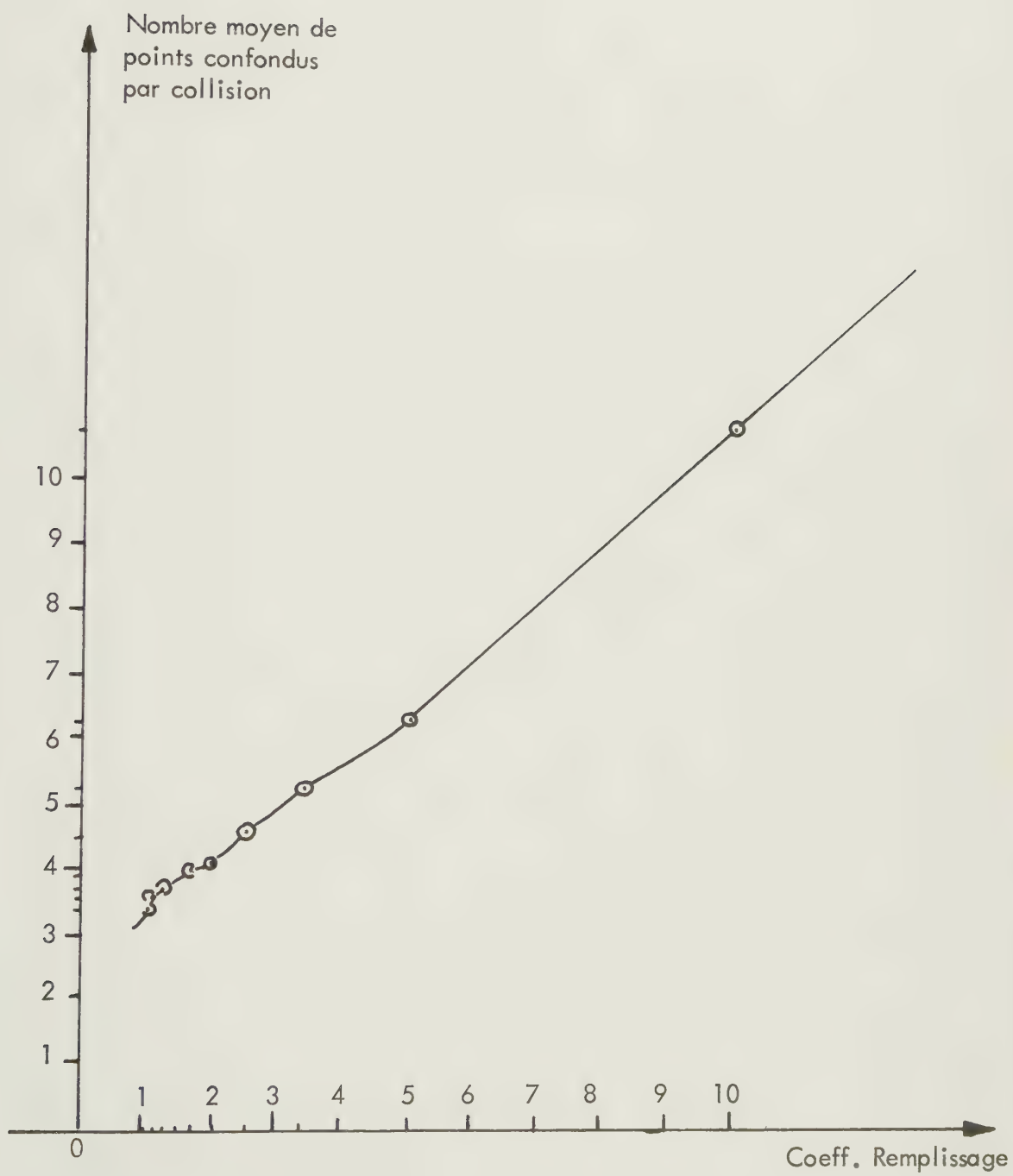
Ces deux noms présentent une collision d'adresse, d'où l'idée de saut de caractères pour rechercher les groupements les plus stables.

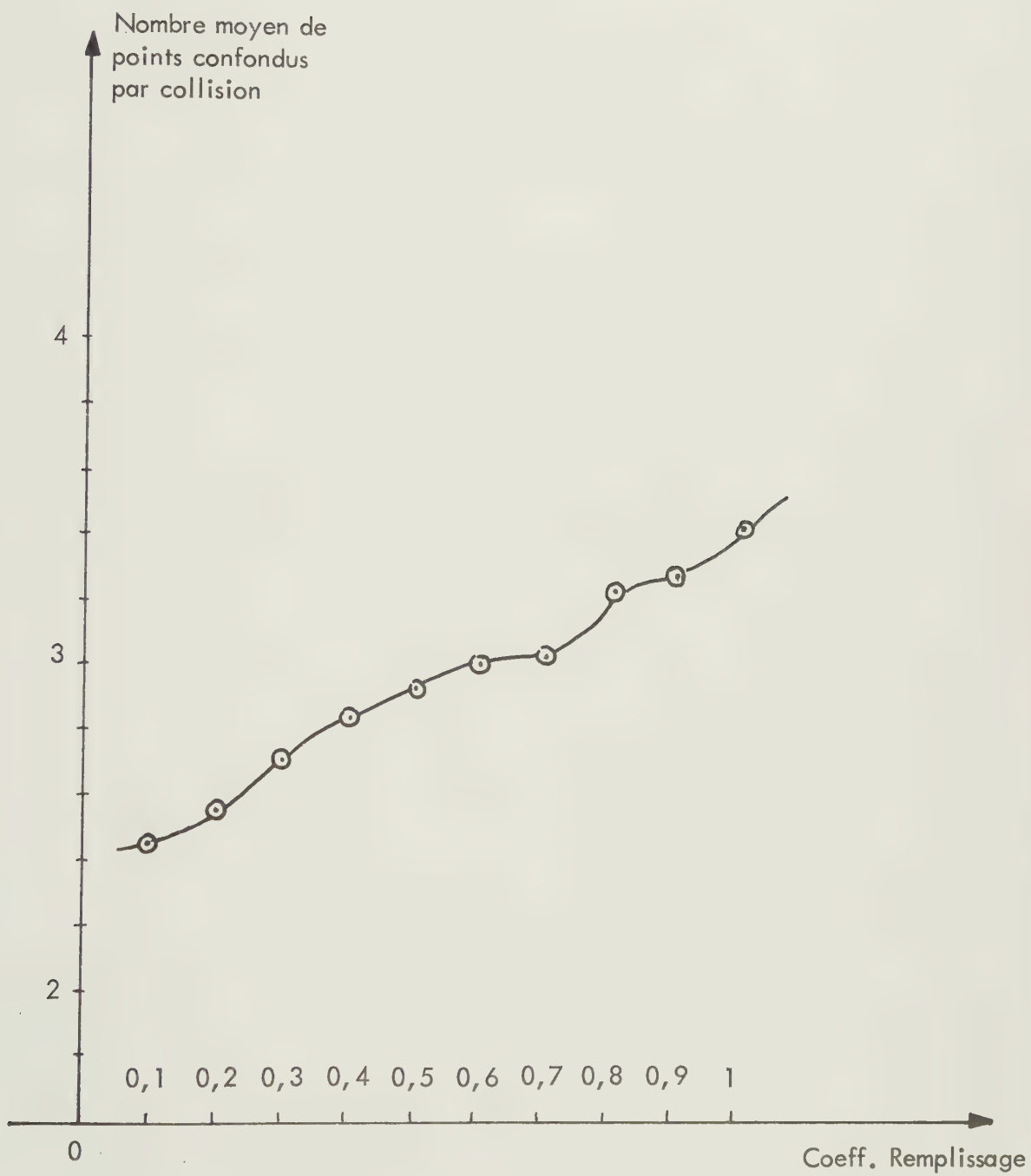
De toute façon, la présence de ces collisions nous a mené à étudier leurs variations en fonction du coefficient de remplissage de la mémoire, en faisant varier ce coefficient de 0,1 en 0,1 (voir tableau de Variation, planche 10).

Les courbes représentatives montrent que lorsque le coefficient de remplissage croît, le nombre moyen de points confondus par collision croît jusqu'à avoir un nombre égal à la taille du fichier lui-même.

Enfin, un dernier avantage de cette méthode est qu'elle conserve l'ordre alphabétique, elle est même conçue en fonction de cet ordre, d'où la possibilité de pratiquer sur le fichier une dichotomie locale.

De plus, comme l'erreur moyenne ne dépasse pas 10 valeurs, une façon pratique d'exploiter ces résultats, c'est l'adressage indexé par paquet de 15 ou 20 noms, avec des renvois à d'autres paquets au cas où il y a dépassement de capacité.





ANNEXE

BIBLIOGRAPHIE

- 1 - C. T. ABRAHAM, S. P. GHOSH, D. K. RAY SHAUDHURY
File Organisation Schemes Based on Finite Geometries
(INFORMATION and CONTROL)
Vol. 12, N° 2, Février 1968
- 2 - P. DEBRAINE
Machines de Traitement de l'Information
(MASSON)
Tomes I, II, III
- 3 - P. FOUCHE
Traité de Prononciation Française
(EDITIONS KLINCKSIECK)
- 4 - J. M. FOSTER
List Processing
(MAC DONALD COMPUTER MONOGRAPHS)
- 5 - G. GUIHO
Thèse Faculté des Sciences PARIS
Représentation et traitement de surfaces en ordinateur (Avril 1970)
- 6 - Y. HARRAND
Traitement des files et des listes
(DUNOD)
- 7 - KAISER - COULON
Thèse Faculté des Sciences PARIS
Conception et réalisation d'un programme d'enseignement assisté (Juin 1970)

- 8 - R. MOREAU
Décomposition en syllabes par ordinateur d'un mot écrit
(Service Développement Scientifique IBM)
Etude N° 1
- 9 - J. C. SIMON
Institut de Programmation (1970) PARIS
Prise et Conservation automatique de l'information - Filtrages
- 10 - Melle. H. VALABREGUE
Transcription phonétique par ordinateur de mots français, et génération
automatique de noms de marque
Etude N° 73
Service Développement Scientifique IBM
- 11 - COMPTE RENDU ACADEMIE DE SCIENCES
2 Novembre 1970
- 12 - HOUDAS, HORVAS
Thèse Faculté des Sciences PARIS Juin 1970
- 13 - C. PICARD
Problèmes de recherche de l'information et questionnaires
Congrès AFIRO Versailles 1964
(DUNOD 1965)
- 14 - IVAN FLORES
Computer Sorting
(Prentice & Hall 1969)
- 15 - C. SHANNON
A mathematical theory of communication
(BELL System Technology)
Vol. 27, N° 7, 1948
UNIVERSITY OF ILLINOIS Press.

PLANCHES

- 1 - Etude phonétique pure
- 2 - Découpage en syllabes grammaticales
- 3 - Découpage en syllabes phonétiques
- 4 - Comptage
- 5 - Variation du code résultant partie droite -
demi-octet par demi-octet
- 5.1 - Variation du code résultant partie gauche -
demi-octet par demi-octet
- 6 - Calcul d'adresses
- 7 - Utilisation des listes enchaînées
- 8 - Matrice des probabilités isolées
- 9 - Matrice des probabilités cumulées
- 10 - Tableau de variation du nombre moyen de points par
collision

PLANCHE 1

RICHET
 ZENOU
 CAZAUX
 ROCHER
 DUTHERAT
 FERRE
 EDOUARD
 BOUEYGUET
 DUFOUR
 EVE
 TOUSSAINT
 VILLE
 MORVAN
 RABAT
 REGNER
 REINARD
 REINHOLD
 REINHARD
 BAYARD
 AIFUL
 QUILLON
 WILLIAMS
 ESFLAIN
 ESTEREL
 ESTAINVILLE
 VIER
 VIERON
 VIEILLARD
 VIELLARD
 QUILLON
 ESTABLIE
 MOREAU
 ADRIEN
 AILLAUD
 ARINAL
 BEAUGAS
 BRICOUT
 CAFFOZ
 DAUPHIN
 DEMARNE
 DORNBUSCH
 EVRARD
 FORCADE
 GOJOT
 IKIZEK
 ISMAN
 KUGALNICEANU
 LACRONIQUE
 LUNARDELLI
 MARLAUD
 MARTIN
 MATHIEU
 MIARA
 MICHAUD
 PAGES
 PARMENTIER
 PIOCH
 ROLLAND
 SALES
 SOMIA
 VALABREGUE

RICHAR
 ZENOU
 KAZOU
 ROCHER
 DUTHERAT
 FERRE
 EDOUARD
 BOUEYGUET
 DUFOUR
 EVE
 TOUSSAINT
 VILLE
 MORVAN
 RABAT
 REGNER
 REINARD
 REINHOLD
 REINHARD
 BAYARD
 AIFUL
 QUILLON
 WILLIAMS
 ESFLAIN
 ESTEREL
 ESTAINVILLE
 VIER
 VIERON
 VIEILLARD
 VIELLARD
 QUILLON
 ESTABLIE
 MOREAU
 ADRIEN
 AILLAUD
 ARINAL
 BEAUGAS
 BRICOUT
 CAFFOZ
 DAUPHIN
 DEMARNE
 DORNBUSCH
 EVRARD
 FORCADE
 GOJOT
 IKIZEK
 ISMAN
 KUGALNICEANU
 LACRONIQUE
 LUNARDELLI
 MARLAUD
 MARTIN
 MATHIEU
 MIARA
 MICHAUD
 PAGES
 PARMENTIER
 PIOCH
 ROLLAND
 SALES
 SOMIA
 VALABREGUE

PLANCHE 2

DESOUTTER
 DESPATURE
 FAVIER
 FAYET
 FAYOT
 FAYOLLE
 FESQUET
 FOUCAULT
 GALLOT
 GALLOU
 GIBQUIM
 GIBOULCT
 GRANVILLE
 MARESTAING
 MARTEAU
 NICOD
 NEIMON
 NEISSON
 NACAUD
 DERIBREUX
 NACAY
 NACAU
 PIERRE
 PIESSEAU
 PIERRAT
 GUICHARD
 PONCELET
 POMPON
 GUILLEMET
 GUILLARD
 RONDELEUX
 GILBERT
 RONGET
 HENRI
 SAMBURCY
 SAMOIL
 SAVOYE
 HOLLARD
 HULOT
 SAVOIE
 IMBAULT
 JALLON
 SEE
 SEYDOUX
 SIMONNET
 AUGIER
 SIMONNEAU
 SIMONOD
 SICNEAU
 SPIGUEL
 STEINER
 STRAUSS
 SUCOTRA
 SUCHET
 SUISSA
 BARROIS
 BARROS
 BAVAY
 THEBAUD
 TOUSSAINT
 VALLÉE

DESOUTTER
 DESPAUTURE
 FAVIER
 FAYET
 FAYOT
 FAYOLLE
 FESQUET
 FOUCAULT
 GALLOT
 GALLOU
 GIBQUIM
 GIBOULCT
 GRANVILLE
 MARESTAING
 MARTEAU
 NICOD
 NEIMON
 NEISSON
 NACAUD
 DERIBREUX
 NACAY
 NACAU
 PIERRE
 PIESSEAU
 PIERRAT
 GUICHARD
 PONCELET
 POMPON
 GUILLEMET
 GUILLARD
 RONDELEUX
 GILBERT
 RONGET
 HENRI
 SAMBURCY
 SAMOIL
 SAVOYE
 HOLLARD
 HULOT
 SAVOIE
 IMBAULT
 JALLON
 SEE
 SEYDOUX
 SIMONNET
 AUGIER
 SIMONNEAU
 SIMONOD
 SICNEAU
 SPIGUEL
 STEINER
 STRAUSS
 SUCOTRA
 SUCHET
 SUISSA
 BARROIS
 BARROS
 BAVAY
 THEBAUD
 TOUSSAINT
 VALLEE

PLANCHE 3

VALARFECHE
 WEL SCH
 EEMONT
 SAVIGNY
 SCHAAL
 SECRET
 LEGRANDCIS
 BERNARD
 ELENFAIT
 BESNIER
 EERTHE
 EOUARD
 CHEVALIER
 CLEROC
 CLERDOUET
 COISY
 COKHAN
 CICTER
 DIDELIN
 DETILLFUX
 DESOUTTER
 DESPATRE
 FAVIER
 FAYET
 FAYET
 FAYOLLE
 FESQUET
 FOUCAULT
 GALLOT
 GALLOU
 GIECUIN
 GIBELLOTT
 GRANVILLE
 MARCSTAING
 MARTEAU
 NICCUE
 NEMON
 NEISSON
 NADAU
 DEFFIEDEUX
 NADAY
 NADAU
 PIERRE
 PIESSEAL
 PIERRAT
 GUICHARD
 PONCELOT
 POMFON
 GUILLEMET
 GUILLARD
 RONDELUX
 GILBERT
 RONGET
 HENRI
 SAMBURY
 SAMOIL
 SAVOYE
 HOLLARD
 HULLOT
 SAVOIE

VA LA BRE GUE
 MEL SCH
 BE MONT
 SA VI NI YI
 SCHAAL
 SE CRET
 LE GRAN DUIS
 BER NAR D
 BI YEN FAIT
 BE SNI YER
 BER THE
 BOUAR D
 CHE VA LI YER
 CLER CO
 CLER DOUET
 DUIS
 DO KHAN
 DI DI YER
 DI DE LIN
 DE TI YFUX
 DE SOL TTER
 DE SPA TU RE
 FA VI YER
 FA YET
 FA YOT
 FA YO LLE
 FE SKET
 FOU CAUL T
 GA LLOT
 GA LLOU
 JI BOLIM
 JI BOU LOT
 GRAN VI YE
 MA RE STAIN G
 MAR TEAU
 NI COUD
 NEI MON
 NEI SSON
 NA DAUD
 DE RI BREUX
 NA DAI
 NA DAU
 PI YE RRE
 PI YE SSEAU
 PI YE RRAT
 GUI CHAR D
 PON SE LET
 POM PON
 GUI YE MET
 GUI YAR D
 RON DE LEUX
 JIL BER T
 PON JET
 HEN RI
 SAM BUR SI
 SA MOY
 SA VOI YE
 HO LLAR D
 HU LOT
 SA VO YE

3 SUITE

IMBAULT
 JALLON
 SEE
 SEYDOUX
 SIMONNET
 AUCIER
 SIMONNEAL
 SIMONOD
 SIGNATAU
 SPIGUEL
 STEINER
 STRAUSS
 SUCOTEA
 SUCHET
 SUSSA
 EAFROIS
 EAFROS
 BAVAY
 THEBAUD
 TOLSSAINT
 VALLEE
 VALLET
 VAYLET
 VALX
 VICAL
 VICTORIA
 VENTURA
 VENNIN
 VIDILLES
 VILLEGRE
 VILLAUDRE
 VILLAIN
 VIVIEN
 WEEER
 AUFFRAY
 TOURNIER
 ROUSSEL
 YOUNG
 SCHLUMBERGER
 LELCUP
 YOLSHKO
 GUTTSCHNEIDER
 CHRISMAN
 REICHMAN
 GUIHO
 ARAGLE
 ARNAUDIN
 LEBODRE
 COMICT
 LAZARD
 LAYANI
 LAVRILLIET
 LAVRILUX
 LAVILLE
 LATOURTE
 LAUNAY
 LANGLOIS
 LANTZ
 LAISSY
 LAISSUS

IMBAULT
 JALLON
 SEE
 SEYDOUX
 SIMONNET
 AUJY
 SIMONNEAU
 SIMONOD
 SIGNATAU
 SPIGUEL
 STEINER
 STRAUSS
 SUCOTEA
 SUCHET
 SUSSA
 EAFROIS
 EAFROS
 EAVAI
 THEBAUD
 TOLSSAINT
 VALLEE
 VALLET
 VAYLET
 VAUX
 VICAL
 VICTORIA
 VENTURA
 VENNIN
 VIDILLES
 VILLEGRE
 VILLAUDRE
 VILLAIN
 VIVIEN
 WEEER
 AUFFRAY
 TOURNIER
 ROUSSEL
 YOUNG
 SCHLUMBERGER
 LELCUP
 YOLSHKO
 GUTTSCHNEIDER
 CHRISMAN
 REICHMAN
 GUIHO
 ARAGLE
 ARNAUDIN
 LEBODRE
 COMICT
 LAZARD
 LAYANI
 LAVRILLIET
 LAVRILUX
 LAVILLE
 LATOURTE
 LAUNAY
 LANGLOIS
 LANTZ
 LAISSY
 LAISSUS

PLANCHE 4

	TOTAL	PRINC.	MODIF.
A	0000000000001555	0000000000000000	0000000000000000
B	0000000000000385	0000000000000379	0000000000000006
C	0000000000000169	0000000000000127	0000000000000042
D	0000000000000486	0000000000000342	0000000000000144
E	0000000000001450	0000000000000000	0000000000000000
F	0000000000000083	0000000000000072	0000000000000011
G	0000000000000190	0000000000000167	0000000000000023
H	0000000000000000	0000000000000000	0000000000000000
I	0000000000001395	0000000000000000	0000000000000336
J	0000000000000167	0000000000000166	0000000000000001
K	0000000000000114	0000000000000081	0000000000000020
L	0000000000000560	0000000000000391	0000000000000169
M	0000000000000332	0000000000000336	0000000000000056
N	0000000000000926	0000000000000316	0000000000000610
O	00000000000001040	0000000000000000	0000000000000000
P	0000000000000191	0000000000000180	0000000000000011
Q	0000000000000004	0000000000000000	0000000000000004
R	00000000000001046	0000000000000040	0000000000000646
S	0000000000000363	0000000000000206	0000000000000157
T	0000000000000566	0000000000000236	0000000000000330
U	0000000000000726	0000000000000000	0000000000000000
V	0000000000000216	0000000000000215	0000000000000001
W	0000000000000025	0000000000000022	0000000000000003
X	0000000000000054	0000000000000006	0000000000000048
Y	0000000000000548	0000000000000231	0000000000000000
Z	0000000000000078	0000000000000055	0000000000000023
AI	0000000000000146	0000000000000000	0000000000000000
"	0000000000000318	0000000000000000	0000000000000000
BL	0000000000000016	0000000000000016	0000000000000000
BR	0000000000000044	0000000000000044	0000000000000000
CL	0000000000000018	0000000000000018	0000000000000000
CR	0000000000000007	0000000000000007	0000000000000000
CH	0000000000000115	0000000000000110	0000000000000005
CU	0000000000000000	0000000000000000	0000000000000000
CS	0000000000000000	0000000000000000	0000000000000000
CZ	0000000000000006	0000000000000003	0000000000000003
DJ	0000000000000010	0000000000000010	0000000000000000
DL	0000000000000007	0000000000000007	0000000000000000
DR	0000000000000042	0000000000000042	0000000000000000
FL	0000000000000006	0000000000000006	0000000000000000
FR	0000000000000014	0000000000000014	0000000000000000
GL	0000000000000009	0000000000000009	0000000000000000
GR	0000000000000037	0000000000000037	0000000000000000
GN	0000000000000000	0000000000000000	0000000000000000
GU	0000000000000000	0000000000000000	0000000000000000
KH	0000000000000003	0000000000000003	0000000000000000
KL	0000000000000002	0000000000000002	0000000000000000
KR	0000000000000003	0000000000000003	0000000000000000
KS	0000000000000004	0000000000000004	0000000000000000

SUITE 4

PH	000000000000000000	000000000000000000
PL	000000000000000017	000000000000000017
PR	000000000000000022	000000000000000022
PS	000000000000000002	000000000000000002
QH	000000000000000000	000000000000000000
RH	000000000000000002	000000000000000002
TH	000000000000000039	000000000000000039
TR	000000000000000044	000000000000000044
TS	000000000000000003	000000000000000003
TZ	000000000000000012	000000000000000012
VR	000000000000000012	000000000000000012
SC	000000000000000012	000000000000000012
SH	000000000000000002	000000000000000002
LH	000000000000000015	000000000000000015
QU	000000000000000000	000000000000000000
TCH	000000000000000007	000000000000000007
BH	000000000000000001	000000000000000001
CQ	000000000000000000	000000000000000000
CK	000000000000000000	000000000000000000
NH	000000000000000006	000000000000000006
MH	000000000000000006	000000000000000006
BB	000000000000000023	000000000000000023
CC	000000000000000003	000000000000000003
DD	000000000000000006	000000000000000006
FF	000000000000000028	000000000000000028
GG	000000000000000002	000000000000000002
LL	0000000000000000116	0000000000000000116
MM	000000000000000007	000000000000000007
NN	000000000000000070	000000000000000070
PP	000000000000000002	000000000000000002
RR	000000000000000051	000000000000000051
SS	0000000000000000109	0000000000000000109
TT	000000000000000041	000000000000000041
ZZ	000000000000000007	000000000000000007
GR	000000000000000000	000000000000000000
ST	000000000000000051	000000000000000051
SP	000000000000000007	000000000000000007
GH	000000000000000003	000000000000000003
SK	000000000000000018	000000000000000018
DO	000000000000000000	000000000000000000
SL	000000000000000012	000000000000000012
SN	000000000000000017	000000000000000017
SM	000000000000000012	000000000000000012
SHR	000000000000000000	000000000000000000
SCH	000000000000000037	000000000000000037
CHR	000000000000000004	000000000000000004
PHR	000000000000000000	000000000000000000
PHT	000000000000000000	000000000000000000
THR	000000000000000006	000000000000000006
CHL	000000000000000000	000000000000000000
CHM	000000000000000000	000000000000000000
CHN	000000000000000000	000000000000000000
SPR	000000000000000001	000000000000000001
SPL	000000000000000001	000000000000000001
SCL	000000000000000001	000000000000000001
STR	000000000000000006	000000000000000006
TZL	000000000000000003	000000000000000003
SCHN	000000000000000005	000000000000000005
SCHR	000000000000000001	000000000000000001
SCHM	000000000000000003	000000000000000003



AD.VRAIE PLANCHE 6 AD.CALC.

10	2.010	9.018
11	2.044	9.201
12	2.337	10.548
13	4.106	12.920
14	5.210	14.780
15	5.211	14.786
16	5.217	14.820
17	5.278	15.164
18	5.477	16.394
19	5.489	16.469
20	5.985	19.955
21	6.083	20.654
22	6.116	20.890
23	6.278	22.027
24	6.348	22.511
25	6.349	22.518
26	6.650	24.506
27	7.314	27.652
28	7.582	28.342
29	8.121	29.074
30	35.147	30.982
31	35.148	30.990
32	35.157	31.058
33	35.381	32.816
34	35.415	33.104
35	35.524	34.032
36	35.949	37.659
37	35.987	37.966
38	35.989	37.982
39	36.185	38.966
40	36.231	39.166
41	36.483	40.110
42	38.055	41.978
43	38.058	41.979
44	39.423	46.972
45	39.491	47.540
46	39.678	49.098
47	39.692	49.209
48	39.709	49.345
49	39.711	49.362
50	39.745	49.641
51	39.759	49.750
52	40.225	52.281
53	40.597	52.988
54	41.545	54.610
55	41.554	54.628
56	42.601	55.526
57	43.701	61.338
58	43.702	61.350
59	43.768	62.140
60	43.771	62.180
61	43.772	62.193
62	43.818	62.787
63	43.937	64.480
64	43.969	64.964
65	43.970	64.980
66	44.035	66.332
67	44.038	66.384
68	44.039	66.409
69	44.069	67.167

6.1

70	44.093	67.777
71	44.503	78.969
72	44.537	79.899
73	44.538	79.928
74	44.553	80.363
75	44.554	80.392
76	44.557	80.450
77	44.573	80.912
78	44.587	81.286
79	44.588	81.315
80	44.605	81.775
81	44.620	82.204
82	44.654	83.114
83	44.675	83.678
84	44.757	85.866
85	44.789	86.703
86	44.838	87.946
87	44.841	87.998
88	44.854	88.334
89	44.855	88.360
90	44.870	88.720
91	44.871	88.745
92	45.054	92.367
93	45.256	94.194
94	45.826	96.365
95	45.872	96.430
96	45.889	96.452
97	46.077	96.576
98	46.089	97.039
99	46.628	99.596
100	46.958	100.862
101	47.977	101.921
102	48.029	102.257
103	48.066	102.589
104	48.280	104.724
105	48.330	105.260
106	48.345	105.418
107	48.750	109.904
108	48.783	110.263
109	48.801	110.452
110	48.884	111.321
111	48.896	111.437
112	48.900	111.479
113	49.128	113.023
114	49.161	113.154
115	50.083	115.177
116	50.086	115.187
117	50.362	116.137
118	50.884	117.858
119	51.522	118.739
120	52.272	120.729
121	52.505	122.014
122	52.530	122.171
123	52.538	122.224
124	52.590	122.566
125	52.606	122.672
126	53.057	126.616
127	53.060	126.658
128	53.061	126.672
129	53.124	127.528
130	53.125	127.543

6.2

131	53.140	127.760
132	53.145	127.818
133	53.292	129.903
134	53.323	130.352
135	53.325	130.382
136	53.394	131.371
137	53.608	134.392
138	53.675	135.297
139	53.691	135.505
140	55.695	139.801
141	56.547	142.421
142	56.583	142.609
143	56.807	143.935
144	56.815	143.982
145	56.899	144.545
146	57.301	148.224
147	57.354	148.759
148	57.355	148.769
149	57.384	149.070
150	57.417	149.413
151	57.537	150.645
152	57.620	151.481
153	57.837	153.588
154	57.935	154.474
155	57.985	154.904
156	58.641	157.437
157	58.903	157.966
158	58.916	157.988
159	59.709	159.455
160	59.975	159.979
161	60.809	162.464
162	60.825	162.527
163	60.852	162.630
164	61.583	166.400
165	61.593	166.452
166	61.612	166.556
167	61.732	167.217
168	61.814	167.661
169	62.111	169.023
170	62.180	169.261
171	62.212	169.370
172	63.716	173.483
173	63.898	173.892
174	63.902	173.901
175	65.084	175.575
176	65.318	177.656
177	65.327	177.736
178	65.353	177.990
179	65.377	178.226
180	65.420	178.654
181	65.449	178.950
182	65.903	183.756
183	65.908	183.812
184	65.974	184.501
185	66.004	184.808
186	66.008	184.844
187	66.021	184.952
188	66.223	186.625
189	66.422	188.137
190	66.522	188.837
191	66.539	188.949

6.3

192	67.461	192.420
193	68.062	192.572
194	68.526	191.464
195	68.546	191.499
196	69.347	209.253
197	69.348	209.300
198	69.350	209.394
199	69.469	214.921
200	69.472	215.070
201	69.481	215.506
202	69.485	215.706
203	69.489	215.856
204	69.490	215.906
205	69.497	216.257
206	69.500	216.407
207	69.502	216.508
208	69.515	217.113
209	69.518	217.283
210	69.528	217.739
211	69.530	217.872
212	69.564	219.524
213	69.565	219.556
214	69.567	219.658
215	69.572	219.915
216	69.576	220.089
217	69.581	220.340
218	69.582	220.378
219	69.614	221.981
220	69.623	222.450
221	69.732	227.911
222	69.734	228.015
223	69.736	228.120
224	69.748	228.703
225	69.751	228.861
226	69.756	229.133
227	69.768	229.736
228	69.768	229.762
229	69.769	229.815
230	69.777	230.187
231	69.781	230.399
232	69.785	230.611
233	69.786	230.664
234	69.798	231.248
235	69.801	231.408
236	69.818	232.257
237	69.833	233.021
238	69.839	233.372
239	69.851	233.956
240	69.852	234.010
241	69.853	234.063
242	70.154	248.163
243	70.160	248.429
244	70.267	252.881
245	70.269	252.967
246	70.270	253.010
247	70.271	253.053
248	70.274	253.181
249	70.279	253.395
250	70.283	253.566
251	70.285	253.651
252	70.291	253.880

6.4

253	70.296	254.077
254	70.302	254.347
255	70.309	254.597
256	70.319	255.007
257	70.324	255.218
258	70.332	255.554
259	70.349	256.223
260	70.353	256.363
261	70.360	256.680
262	70.366	256.886
263	70.386	257.685
264	70.417	258.893
265	70.419	258.974
266	70.536	263.391
267	70.563	264.381
268	70.619	266.392
269	70.634	266.903
270	70.641	267.157
271	70.647	267.338
272	70.686	268.693
273	70.693	268.904
274	70.818	272.941
275	70.837	273.516
276	70.850	273.925
277	70.900	275.400
278	70.909	275.640
279	71.610	285.945
280	71.639	286.153
281	71.653	286.258
282	71.655	286.272
283	71.656	286.280
284	71.665	286.338
285	71.716	286.679
286	71.719	286.699
287	71.720	286.705
288	71.906	287.743
289	72.409	290.002
290	72.457	290.257
291	72.491	290.445
292	72.524	290.642
293	72.706	291.925
294	72.708	291.942
295	72.973	294.689
296	73.624	309.785
297	73.633	310.042
298	73.741	313.524
299	73.758	314.084
300	73.761	314.190
301	73.774	314.615
302	73.777	314.722
303	73.783	314.937
304	73.792	315.224
305	73.795	315.333
306	73.796	315.369
307	73.799	315.491
308	73.807	315.768
309	73.808	315.804
310	73.811	315.914
311	73.812	315.950
312	73.817	316.097
313	73.825	316.391

6.5

314	73.827	316.475
315	73.828	316.502
316	73.841	316.947
317	73.844	317.055
318	73.845	317.096
319	73.852	317.373
320	73.854	317.435
321	73.859	317.603
322	73.861	317.697
323	74.008	323.320
324	74.013	323.575
325	74.019	323.890
326	74.028	324.308
327	74.042	325.044
328	74.044	325.147
329	74.046	325.252
330	74.053	325.570
331	74.059	325.889
332	74.094	327.700
333	74.095	327.754
334	74.121	329.103
335	74.125	329.286
336	74.426	345.492
337	74.428	345.606
338	74.542	351.821
339	74.545	351.992
340	74.546	352.048
341	74.551	352.275
342	74.562	352.859
343	74.576	353.635
344	74.577	353.692
345	74.579	353.804
346	74.580	353.861
347	74.584	354.087
348	74.588	354.313
349	74.597	354.764
350	74.601	354.990
351	74.612	355.609
352	74.614	355.721
353	74.617	355.889
354	74.626	356.338
355	74.637	356.954
356	74.643	357.234
357	74.647	357.458
358	74.648	357.514
359	74.652	357.736
360	74.659	358.126
361	74.660	358.182
362	74.662	358.293
363	74.663	358.349
364	74.668	358.571
365	74.673	358.845
366	74.696	360.067
367	74.857	368.334
368	74.894	370.188
369	74.913	371.101
370	74.920	371.422
371	74.927	371.755
372	74.961	373.392
373	74.963	373.490
374	74.965	373.784

375	74.980	374.270
376	75.086	378.083
377	75.094	378.316
378	75.102	378.585
379	75.113	378.942
380	75.177	380.869
381	75.180	380.961
382	75.187	381.147
383	75.915	395.168
384	75.930	395.318
385	75.932	395.339
386	75.933	395.350
387	75.934	395.360
388	75.938	395.402
389	75.942	395.443
390	75.965	395.663
391	75.979	395.788
392	75.988	395.872
393	75.996	395.936
394	75.999	395.964
395	76.000	395.973
396	76.238	397.091
397	103.571	403.409
398	103.695	405.404
399	103.712	405.689
400	103.731	406.031
401	103.748	406.324
402	103.759	406.509
403	103.778	406.864
404	103.809	407.416
405	103.828	407.785
406	103.832	407.844
407	103.834	407.882
408	103.837	407.941
409	103.838	407.961
410	103.967	410.512
411	103.992	411.030
412	104.032	412.035
413	104.047	412.426
414	104.049	412.482
415	104.071	413.072
416	104.366	421.401
417	104.497	425.167
418	104.498	425.197
419	104.500	425.257
420	104.502	425.317
421	104.505	425.411
422	104.506	425.438
423	104.535	426.277
424	104.548	426.636
425	104.550	426.696
426	104.552	426.755
427	104.564	427.084
428	104.577	427.471
429	104.585	427.709
430	104.594	427.946
431	104.634	429.068
432	104.635	429.215
433	104.640	429.244
434	104.798	433.591
435	104.801	433.647

1473	194.365	1474.718
1474	194.397	1475.109
1475	194.449	1475.744
1476	194.628	1477.927
1477	194.716	1478.976
1478	194.845	1480.452
1479	194.848	1480.477
1480	194.862	1480.637
1481	194.946	1481.526
1482	194.951	1481.578
1483	195.668	1484.673
1484	195.683	1484.707
1485	195.747	1484.852
1486	196.487	1486.794
1487	196.736	1487.695
1488	196.786	1487.897
1489	197.054	1489.111
1490	197.652	1493.008
1491	197.653	1493.014
1492	197.790	1494.287
1493	197.824	1494.624
1494	197.836	1494.750
1495	197.890	1495.328
1496	197.920	1495.659
1497	198.057	1497.597
1498	198.103	1498.425
1499	198.141	1499.125
1500	198.454	1505.176
1501	198.458	1505.266
1502	198.571	1507.458
1503	198.571	1507.469
1504	198.575	1507.546
1505	198.587	1507.781
1506	198.592	1507.873
1507	198.642	1508.834
1508	198.671	1509.379
1509	198.674	1509.445
1510	198.688	1509.692
1511	198.689	1509.712
1512	198.893	1513.369
1513	198.956	1514.413
1514	198.989	1514.929
1515	198.989	1514.940
1516	199.106	1516.021
1517	199.122	1516.143
1518	199.123	1516.150
1519	199.945	1519.004
1520	201.930	1521.767
1521	201.932	1521.786
1522	201.933	1521.801
1523	202.045	1523.753
1524	202.048	1523.835
1525	202.050	1523.890
1526	202.062	1524.213
1527	202.067	1524.336
1528	202.084	1524.789
1529	202.096	1525.124
1530	202.100	1525.248
1531	202.117	1525.714
1532	202.157	1526.863
1533	202.200	1528.126

1534	202.201	1528.129
1535	202.316	1531.671
1536	202.318	1531.710
1537	202.334	1532.236
1538	202.351	1532.766
1539	202.366	1533.247
1540	202.368	1533.298
1541	202.385	1533.850
1542	202.401	1534.370
1543	202.417	1534.870
1544	202.418	1534.910
1545	202.859	1549.128
1546	202.864	1549.289
1547	202.869	1549.428
1548	202.873	1549.542
1549	202.876	1549.643
1550	202.898	1550.298
1551	202.902	1550.432
1552	202.919	1550.927
1553	202.952	1551.901
1554	203.151	1555.549
1555	203.181	1555.945
1556	203.202	1556.208
1557	203.216	1556.394
1558	203.268	1557.011
1559	203.403	1558.400
1560	203.419	1558.540
1561	203.487	1559.113
1562	204.234	1562.675
1563	204.302	1562.911
1564	204.485	1563.445
1565	206.210	1566.840
1566	206.339	1568.212
1567	206.343	1568.258
1568	206.359	1568.432
1569	206.378	1568.644
1570	206.392	1568.803
1571	206.417	1569.085
1572	206.434	1569.275
1573	206.477	1569.770
1574	206.481	1569.813
1575	206.626	1571.490
1576	207.142	1576.667
1577	207.224	1577.171
1578	207.226	1577.179
1579	207.664	1579.284
1580	207.714	1579.467
1581	208.511	1581.585
1582	208.547	1581.648
1583	210.605	1585.388
1584	210.633	1585.531
1585	210.649	1585.617
1586	210.671	1585.727
1587	210.699	1585.875
1588	211.421	1589.210
1589	211.455	1589.339
1590	211.685	1590.148
1591	211.820	1590.554
1592	241.992	1592.000
1593	244.701	1595.511
1594	244.818	1596.069

1595	244.835	1596.147
1596	244.915	1596.504
1597	245.101	1597.038
1598	247.272	1598.055
1599	247.275	1598.057
1600	248.974	1601.828
1601	248.977	1601.850
1602	249.091	1603.040
1603	249.109	1603.252
1604	249.119	1603.375
1605	249.128	1603.492
1606	249.142	1603.656
1607	249.143	1603.673
1608	249.207	1604.476
1609	249.446	1607.655
1610	249.779	1612.242
1611	249.781	1612.273
1612	249.787	1612.359
1613	249.909	1613.968
1614	249.948	1614.457
1615	249.997	1615.073
1616	250.010	1615.196
1617	250.015	1615.233
1618	250.322	1617.553
1619	250.448	1618.321
1620	250.547	1618.847
1621	251.568	1621.221
1622	253.390	1622.854
1623	253.406	1622.914
1624	253.437	1623.031
1625	253.472	1623.167
1626	254.053	1626.378
1627	254.057	1626.418
1628	254.170	1627.485
1629	254.187	1627.649
1630	254.278	1628.546
1631	254.291	1628.679
1632	254.458	1630.330
1633	254.508	1630.823
1634	254.590	1631.612
1635	254.722	1632.844
1636	257.527	1635.880
1637	258.500	1638.165
1638	258.865	1639.530
1639	258.867	1639.538
1640	259.032	1640.015
1641	261.955	1642.896
1642	261.955	1642.899
1643	261.972	1642.952
1644	262.745	1646.055
1645	262.789	1646.232
1646	262.794	1646.253
1647	262.993	1647.016
1648	263.149	1647.485
1649	263.346	1648.009
1650	266.219	1650.465
1651	266.266	1650.596
1652	266.270	1650.608
1653	267.005	1653.007
1654	267.067	1653.197
1655	267.572	1654.633

6.10

1656	266.453	1656.031
1657	270.461	1658.824
1658	270.462	1658.829
1659	270.580	1659.228
1660	270.634	1659.471
1661	271.615	1661.210
1662	302.583	1662.000
1663	318.375	1662.981
1664	322.602	1663.986
1665	334.683	1664.981
1666	335.443	1665.551
1667	339.722	1667.165
1668	339.288	1667.240
1669	400.000	1669.000
1660	0.165	4.552

FICH.1

PLANCHE 7

305	REIZINE	0
306	ROYER	0
307	ROYANT	0
308	ROYON	0
309	RUDAUT	0
310	ROITEL	0
311	RUOSSEL	0
312	RUSKONE	0
313	RICU	0
314	RIBOT	233
315	RIBET	0
316	RITTER	0
317	RIVET	0
318	RICHARD	234
319	RICHET	235
320	RIGAUT	0
321	RICQUARD	0
322	RAMBOURG	0
323	ROMBI	0
324	RONDELEUX	0
325	RONGET	0
326	REINBOULD	0
327	MATTHIS	0
328	MAILLARD	237
329	MAILLOT	238
330	MAYETTE	240
331	MALHCUITRE	0
332	MARLAND	0
333	MARIE	0
334	MARNEAU	0
335	MAGNE	0
336	MATBA	245
337	MABEIX	0
338	MATHCU	246
339	MATHET	249
340	MARTIE	250
341	MARTAIN	252
342	MARTUN	0
343	MARTIN	253
344	NAERVAL	0
345	MARVAN	0
346	MARVON	0
347	MARVEN	257
348	MARCHAL	0
349	MAGEOTTE	0
350	MARCE	0
351	MARCILLE	0
352	MASSON	0
353	MASSIN	0
354	MAQUAIR	259
355	MAGUY	260
356	NARCON	0
357	MARGUIN	0
358	MOLAS	0
359	MOLLET	261
360	MORLON	0
361	MOULIN	0
362	MAURAL	264
363	MORAU	265
364	NORREY	0
365	MAURY	0

FICH.1

SUITE 7.1

366	MORAND	0
367	MAURIN	0
368	MONA	266
369	MCNNOT	0
370	MONNET	0
371	MEONI	0
372	NORMAND	0
373	MONIN	0
374	MCRBARD	0
375	NOBEL	0
376	MORDEL	0
377	MAUDUIT	267
378	NAUDIN	0
379	MAUVAIS	0
380	MORVAN	271
381	MOUSSAOUT	272
382	MCRCEAU	0
383	MAUCCRPS	273
384	NOGUES	0
385	MEYER	0
386	MENAGE	0
387	MAYNGT	274
388	MENES	0
389	NEIMON	0
390	METAY	0
391	NEUVILLE	0
392	MESLARD	0
393	MESSET	0
394	MESNIL	0
395	NEISSON	0
396	MEURGUE	0
397	MOYAT	0
398	MULLER	0
399	MCYCN	0
400	NCIREZ	0
401	MOYRAND	0
402	MCYNAT	275
403	MCYNET	0
404	MOITY	0
405	NUOVO	0
406	MILLAT	0
407	MIOT	276
408	MYLLES	277
409	MILLON	279
410	MINET	0
411	MITRESSE	0
412	NIVEAU	280
413	MICHAUD	281
414	MICHEL	0
415	MICILE	0
416	NIZAN	0
417	NICOUD	0
418	MILGRAM	0
419	MANTEL	282
420	MANGIN	0
421	MANSUY	0
422	MCNTET	0
423	BAYARD	283
424	BAILLCT	284
425	BALLAIZE	286
426	BALUA	0

FICH.2

SUITE 7.2

207	LEFEVRE	0
208	LECOQ	209
209	LEGRUCX	0
210	LECLERC	0
211	LYON	0
212	LANGDOO	0
213	LANDETE	0
214	LANGLET	215
215	LANGUE	0
216	LANGLOIS	217
217	LANGLOIS	0
218	RAMAS	0
219	RAFFIN	0
220	ROUILLARD	0
221	ROUCHES	0
222	ROUSSEL	223
223	ROUSSET	0
224	RENARD	225
225	REINARD	226
226	REINHARD	0
227	RENAUD	228
228	RENCU	229
229	RENOULT	230
230	RENAUD	231
231	RENAULT	0
232	REVEREAU	0
233	RIBOU	0
234	RICHARD	0
235	RICHEZ	236
236	RICHER	0
237	MAILLARD	0
238	MAYOLLE	239
239	MARLAUD	0
240	MAYER	241
241	MAYEUR	242
242	MAYETTE	243
243	MAILLETTE	244
244	MALET	0
245	NACHBA	0
246	NADAU	247
247	NADAUC	248
248	MARTEAU	0
249	NADAY	0
250	MARTY	251
251	MARTY	0
252	MARTEIN	0
253	MARTIN	254
254	MARTIN	255
255	MARTIN	256
256	MARTIN	0
257	MARVENT	258
258	MARFAIN	0
259	MARGUERAY	0
260	MAGGY	0
261	MOLLET	262
262	MOLAY	263
263	MOLIS	0
264	MOURAS	0
265	MOREAU	0
266	MONAT	0
267	MAUDUIT	268

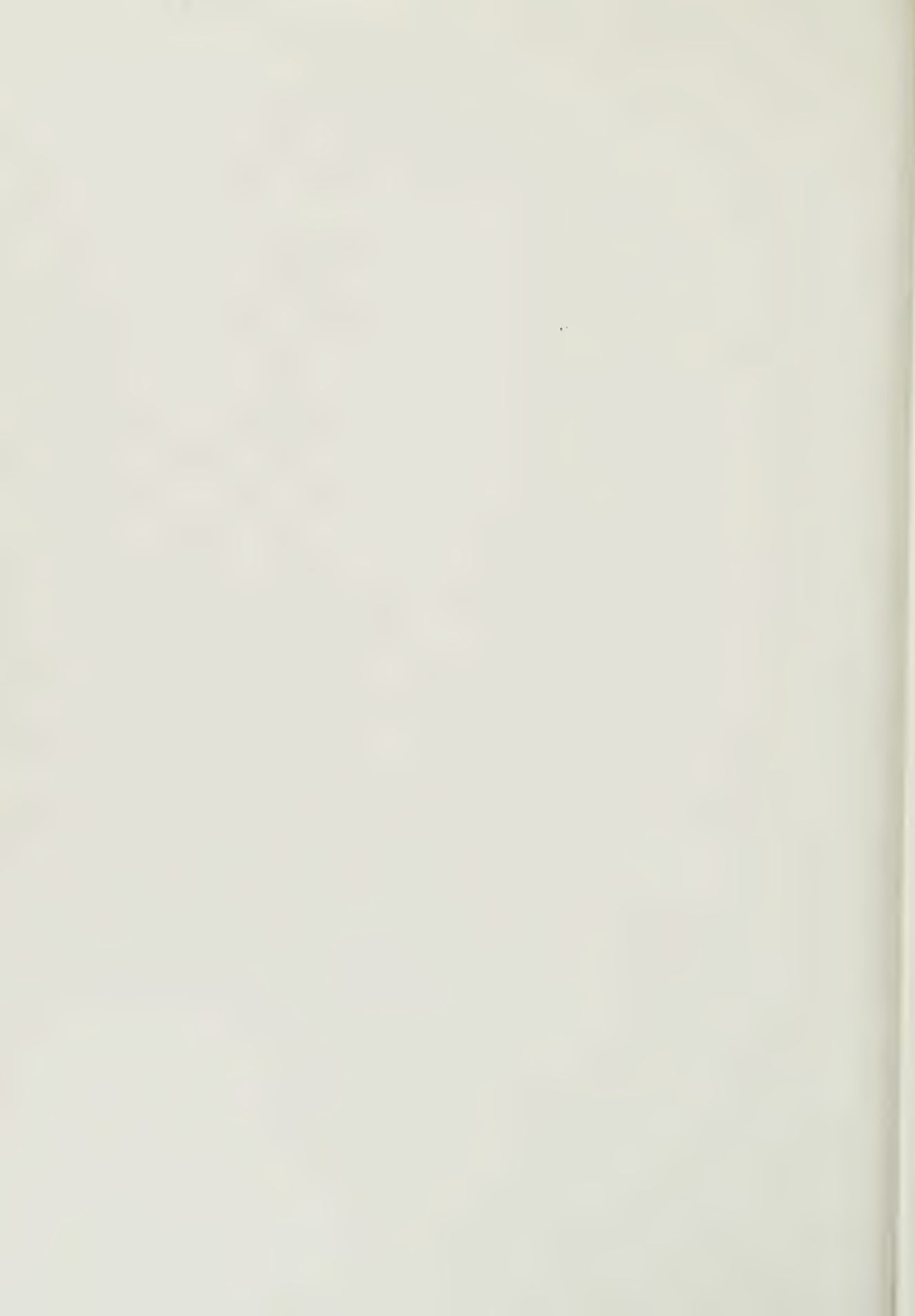
FICH.2

SUITE 7.3

268	MOUTY	269
269	MOATI	270
270	MOATTI	0
271	MORVAN	0
272	MCUSSAOU	0
273	MCQUERO	0
274	MAINAULT	0
275	MOINARD	0
276	MILLCT	0
277	MILER	278
278	MYLLES	0
279	NYONS	0
280	NIVault	0
281	MICHEAU	0
282	NANTEUIL	0
283	PEARLAC	0
284	PAILLGUX	285
285	PAYOT	0
286	BALLEYE	287
287	BAYEUX	288
288	PAYET	289
289	BARTLETT	290
290	BARLET	0
291	BAILLY	0
292	PAYEN	0
293	BAROU	294
294	BARROS	0
295	PARIS	0
296	PEARON	0
297	BARENNE	0
298	BARBE	0
299	PRADC	300
300	PRADEAU	0
301	BARDEZ	302
302	BARDE	0
303	PAGES	0
304	PARSAC	0
305	PLAZER	306
306	PLACET	0
307	BAZIN	308
308	BASQUIN	0
309	BACOT	0
310	PASTEUR	0
311	BOULET	312
312	POULET	313
313	BORLET	0
314	BAURET	315
315	BOUREL	0
316	BROUMET	0
317	BORDERON	0
318	BOUCHER	319
319	BOUCHER	320
320	BOUCHER	321
321	BROCHEC	322
322	BOURGE	0
323	BOSTEL	0
324	BOCCARD	0
325	PORQUET	0
326	PELLET	327
327	PERLET	0
328	PERALLE	329







DEUXIÈME THÈSE

Propositions données par la Faculté

Vu et approuvé

Paris, le 22 DEC. 1970

Le Doyen de la Faculté des Sciences
de Paris

Luc. GAUTHIER

Vu
et permis d'imprimer

Le Recteur de l'Académie de Paris

Robert MALLET

